LabVIEW™

Advanced Signal Processing Toolkit

Time Series Analysis Tools User Manual

August 2005
371532A-01

NATIONAL INSTRUMENTS™

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

Worldwide Offices

Australia 1800 300 800, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599, Canada 800 433 3488, China 86 21 6555 7838, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, India 91 80 51190000, Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400, Lebanon 961 0 1 33 28 28, Malaysia 1800 887710, Mexico 01 800 010 0793, Netherlands 31 0 348 433 466, New Zealand 0800 553 322, Norway 47 0 66 90 76 60, Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 095 783 68 51, Singapore 1800 226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 02 2377 2222, Thailand 662 278 6777, United Kingdom 44 0 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on National Instruments documentation, refer to the National Instruments Web site at ni.com/info and enter the info code feedback.

© 2005 National Instruments Corporation. All rights reserved.

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on `ni.com/legal` for more information about National Instruments trademarks.

Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Contents

About This Manual

Conventions ix
Related Documentation x

Chapter 1
Introduction to Time Series Analysis

Time Series Analysis Objectives 1-1
Time Series Analysis Application Areas 1-2
Fault and Failure Diagnosis 1-2
Structural Testing 1-5
Data Mining 1-7
Industrial Measurement 1-8
Model Predictive Control 1-9
Overview of LabVIEW Time Series Analysis Tools 1-10
Time Series Analysis Methods 1-10
Finding Example VIs 1-12
Related Signal Processing Tools 1-12

Chapter 2
Time Series Analysis Concepts and Methods

Concepts of Time Series 2-1
Continuous Time Series and Discrete Time Series 2-1
Time-Ordered Series and Spatial-Ordered Series 2-2
Univariate Time Series and Multivariate Time Series 2-3
Stationary Time Series and Nonstationary Time Series 2-4
Choosing an Appropriate Method 2-5
Time Series Analysis Procedure 2-5

Chapter 3
Acquiring and Preprocessing Time Series

Obtaining a Discrete Time Series 3-1
Factors that Influence a Discrete Time Series 3-2
Preprocessing a Discrete Time Series 3-3
Resampling a Time Series 3-3
Smoothing a Time Series 3-7
Detrending a Time Series 3-9

Chapter 4
Performing Statistical Analysis

Basic Statistical Analysis Methods 4-1
Understanding the Mean and Variance Values 4-1
Understanding the Skewness and Kurtosis Values 4-2
Understanding the Confidence Limits 4-3
Testing the Normal Distribution 4-4
Checking the Stationarity and Whiteness 4-4
Multivariate Statistical Analysis Methods 4-5
Understanding the Covariance Matrix 4-5
Understanding Principal Component Analysis 4-5
Understanding Independent Component Analysis 4-11

Chapter 5
Building Models and Predicting Time Series Values

Building Models 5-1
Building Autoregressive Models 5-2
Building Autoregressive-Moving Average Models 5-7
Building Modal Parametric Models 5-9
Building Stochastic State-Space Models 5-12
Using the Time Series Modeling Express VI 5-13
Predicting Time Series Values 5-16
ARMA Model-Based Prediction 5-16
Exponential Smoothing Prediction 5-18

Chapter 6
Performing Correlation and Spectral Analysis

Correlation Methods 6-1
Power Spectrum Estimation Methods 6-5
Using the Model-Based Methods 6-6
Using the MUSIC Method 6-9
Using the Periodogram Method 6-10
Using the Welch Method 6-10
Bispectrum Estimation Methods 6-12
Cepstrum Estimation Methods 6-12
Estimating the Real Cepstrum of a Time Series 6-13
Estimating the Complex Cepstrum of a Time Series 6-13
Estimating the Power Cepstrum of a Time Series 6-15

Appendix A
References

Appendix B
Technical Support and Professional Services

About This Manual

This manual provides information about time series concepts, time series analysis objectives, time series analysis applications, different types of methods you can perform on a time series, theoretical basis for each type of method, and application examples provided in the LabVIEW Time Series Analysis Tools.

The *Time Series Analysis Tools User Manual* contains six chapters and is arranged as follows:

- Chapter 1, *Introduction to Time Series Analysis*, introduces the objectives and application areas of time series analysis and provides an overview of the Time Series Analysis Tools.
- Chapter 2, *Time Series Analysis Concepts and Methods*, explains the concepts of time series, provides information about how to choose appropriate methods and describes a typical procedure that you can follow when using the methods to analyze a time series.
- Chapter 3, *Acquiring and Preprocessing Time Series*, describes how to obtain discrete time series and how to perform time series inspection and preprocessing with the Time Series Analysis Tools.
- Chapter 4, *Performing Statistical Analysis*, describes the statistical analysis methods that the Time Series Analysis Tools provide. You can process both univariate and multivariate time series with these methods.
- Chapter 5, *Building Models and Predicting Time Series Values*, introduces various models you can build for univariate or multivariate time series. This chapter also discusses the theoretical background of estimating models and describes the relationship between model coefficients and dynamic characteristics of a time series.
- Chapter 6, *Performing Correlation and Spectral Analysis*, introduces the correlation analysis and various spectrum estimation methods that you can use to analyze a time series in the time domain or frequency domain.

Conventions

The following conventions appear in this manual:

»

The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to

	pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.
	This icon denotes a tip, which alerts you to advisory information.
	This icon denotes a note, which alerts you to important information.
bold	Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.
italic	Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text that is a placeholder for a word or value that you must supply.
`monospace`	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.

Related Documentation

The following documents contain information that you might find helpful as you read this manual:

- *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**
- *LabVIEW User Manual*, available as a printed manual in your LabVIEW development system box or by selecting **Start»All Programs»National Instruments»LabVIEW 7.1»Search the LabVIEW Bookshelf**
- *LabVIEW Analysis Concepts*, available by selecting **Start»All Programs»National Instruments»LabVIEW 7.1»Search the LabVIEW Bookshelf**
- *LabVIEW Measurements Manual*, available by selecting **Start»All Programs»National Instruments»LabVIEW 7.1»Search the LabVIEW Bookshelf**

Refer to Appendix A, *References*, for a list of textbooks and technical papers that National Instruments used to develop the Time Series Analysis Tools.

1

Introduction to Time Series Analysis

A time series is a sequence of observed values, for example, the air temperature in meteorological science, blood pressure in biomedical science, or vibration in mechanical engineering or civil engineering. Time series analysis uses a collection of systematic approaches to extract information about the characteristics of a physical system that generates time series. Approaches to time series analysis include estimating statistical parameters, building dynamic models, performing correlations, computing the power spectral density (PSD), and others.

This chapter introduces the objectives and application areas of time series analysis and provides an overview of the LabVIEW Time Series Analysis Tools.

Time Series Analysis Objectives

Time series analysis is useful when you want to extract information from a time series, to discover the characteristics of a physical system that generates the time series, to predict the changes of a time series, or to improve control over the physical system. The objectives of time series analysis are as follows:

- **Description**—You can use the time series analysis methods to obtain descriptive or statistical measures of a time series. For example, to measure the trends or periodicity, you can plot the time series. To measure the symmetry of a time-series amplitude distribution, you can examine a histogram of the amplitude of a time series.
- **Explanation**—You can use the observed variation of a time series to explain the variation of a related time series, which can help you understand the nature of the relationship between the two signals. For example, you can explain the dynamic properties of a physical system by analyzing the input time series to and output time series from the system.

- **Prediction**—You can use observed values to predict the future values of a time series. For example, you can predict the future variation of a time series in an industrial process and make sure the process is working properly.
- **Control**—You can use the predicted values of a time series to determine appropriate corrective actions that you take to specify optimal settings for the controller and keep a physical system or process operating properly.

The Time Series Analysis Tools provide a set of Time Series Analysis VIs that you can use to analyze or process a time series. Refer to the *Overview of LabVIEW Time Series Analysis Tools* section of this chapter for more information about the Time Series Analysis Tools.

Time Series Analysis Application Areas

Time series occur in many fields. This section discusses time series analysis applications in the industrial and engineering fields using the Time Series Analysis Tools.

The Time Series Analysis Tools provide example VIs for different application areas. In the Browse tab of the NI Example Finder, you can view these example VIs by selecting **Toolkits and Modules»Time Series Analysis»Applications**. Refer to the *Finding Example VIs* section of this chapter for information about launching the NI Example Finder.

Fault and Failure Diagnosis

Fault diagnostics are important industrial tools to assess the health of industrial equipment and ensure that the equipment is in proper working condition. Failure or damage detection ensures the integrity of machine elements and structures. Using the time series analysis methods such as dynamic modeling, cepstrum analysis, or bispectrum analysis, you can perform fault and failure diagnosis by analyzing the vibration or acoustic signals from the equipment.

Figure 1-1 shows an example of performing fault diagnosis by building an autoregressive-moving average (ARMA) model for a vibration time series from a running engine.

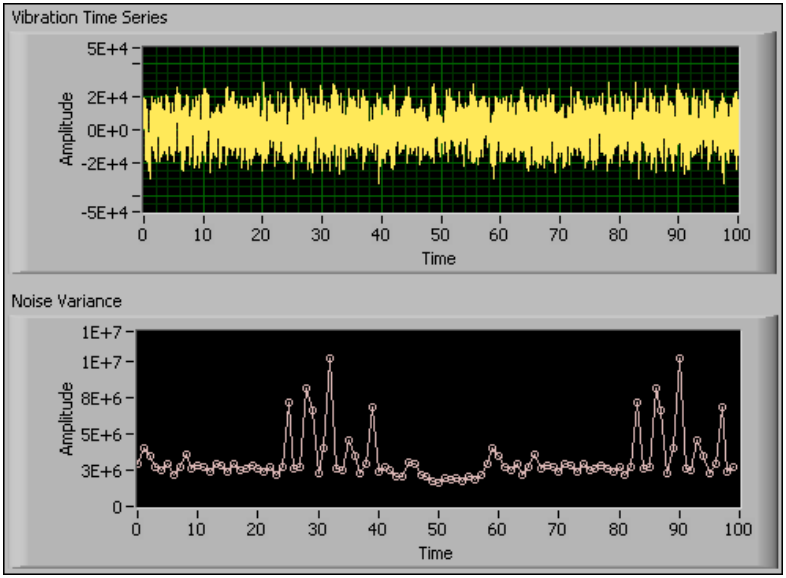

Figure 1-1. Time Series Modeling for Fault Diagnosis

Under normal conditions, the vibration signal from the engine is a stationary time series. If you build an ARMA model for this stationary time series, the modeling errors are usually small. However, if the engine is not running properly due to imbalance or cracks, the vibration signal becomes a nonstationary time series. If you build an ARMA model for this nonstationary time series, the modeling errors increase. In Figure 1-1, the peaks in the **Noise Variance** graph show the large variances of the modeling errors and indicate that this engine is not running properly. Refer to Chapter 5, *Building Models and Predicting Time Series Values*, for information about building ARMA models.

If you want to detect a structural failure or damage in a mechanical system, you usually compute and examine the PSD of the time series that the analyzed system generates. However, in some cases, you cannot get a satisfactory result by computing the PSD. Figure 1-2 shows the PSD of the vibration time series from a normal concrete beam and a cracked concrete beam. The differences are subtle, and they do not suggest the presence of a defect in the beam.

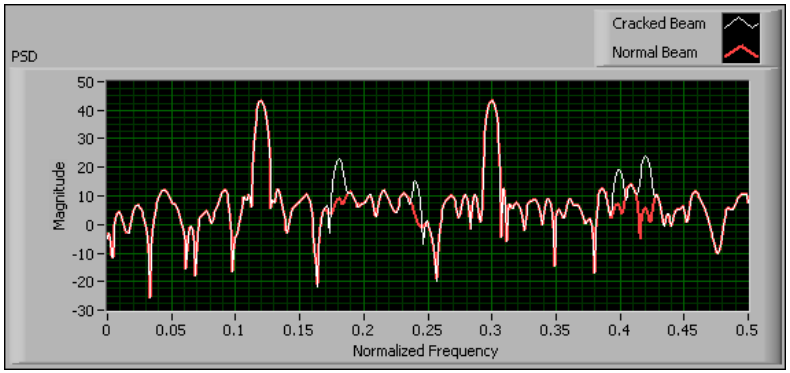

Figure 1-2. Power Spectra of the Cracked Beam and Normal Beam

Using the Time Series Analysis Tools, you can compute the bispectra of the two time series. Bispectrum analysis is related to the third moment (skewness) of a vibration time series and outperforms traditional PSD analysis in detecting the asymmetric nonlinearity due to structural cracks. Refer to Chapter 6, *Performing Correlation and Spectral Analysis*, for information about estimating the power spectrum of a time series.

Figure 1-3 displays the bispectra of the vibration time series from the cracked beam and the normal beam.

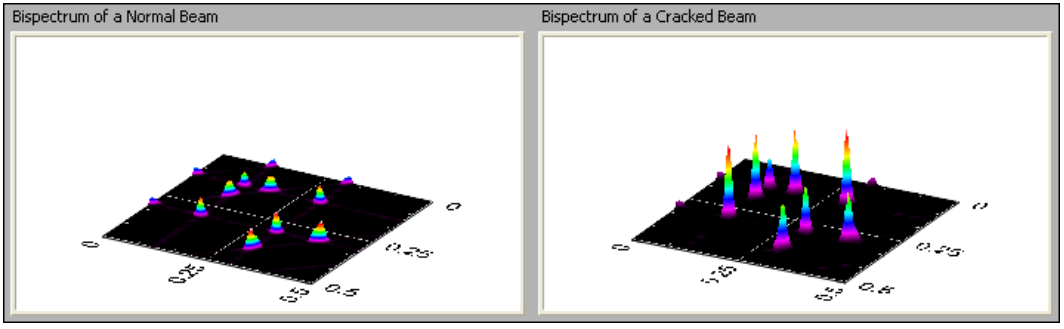

Figure 1-3. Bispectra of the Cracked Beam and Normal Beam

The magnitudes of the peaks in the two bispectra are different. In the **Bispectrum of a Normal Beam** graph, the magnitudes are small. In the **Bispectrum of a Cracked Beam** graph, the magnitudes are large. A large magnitude indicates large coupling between frequencies in a time series. In Figure 1-3, you can see that the bispectrum of the cracked beam contains significant coupling between frequencies due to system non-linearities.

In the NI Example Finder, refer to the Beam Crack Detection VI for more information about performing failure diagnosis with the estimated bispectrum of a time series.

Structural Testing

Structural testing extracts key resonance features of a physical system by estimating the modal parameters of a time series that the system generates. Modal parameters include natural frequencies, damping factors, magnitudes, and phases.

Modal parameters contain information that describes the inherent dynamic properties of a structure. Understanding the vibration behavior of a structure is important in creating robust prototypes and validating structural systems such as cars, aircraft, bridges, and buildings. You can obtain the modal parameters of a structure by performing modal analysis using the time series modeling method.

Figure 1-4 illustrates a structural testing experiment that obtains the modal parameters of a steel-reinforced concrete beam. A hammer impacts the beam, and seven acceleration sensors located in different positions on the beam acquire the resulting vibration signals.

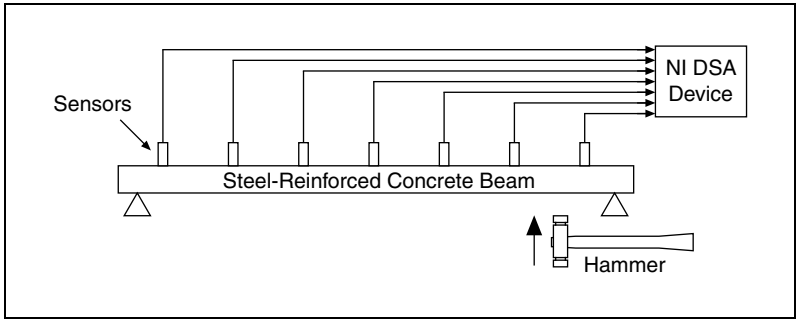

Figure 1-4. Structural Testing of a Steel-Reinforced Concrete Beam

Using the Time Series Analysis Tools, you can compute the resonance components, or modes, of the steel-reinforced concrete beam. Table 1-1 lists the detected natural frequencies f and damping factors α of each mode of the beam.

Table 1-1. Detected Natural Frequencies and Damping Factors

	First Mode	Second Mode	Third Mode	Fourth Mode
f (Hz)	78.2886	249.407	457.382	579.891
α	0.17	0.16	0.17	0.49

Besides the natural frequency and damping factor, a mode also includes the magnitude and phase information. Figure 1-5 shows the modal shapes of the beam computed with the estimated magnitudes and phases of each mode.

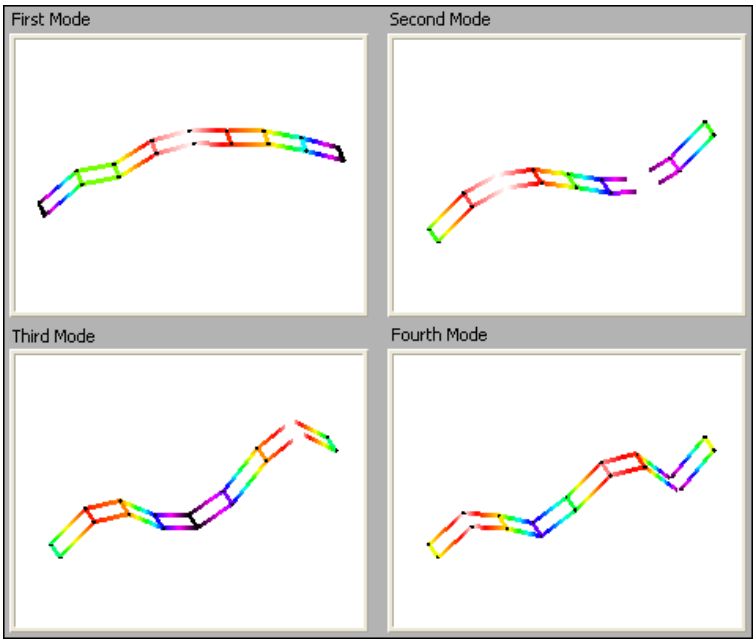

Figure 1-5. Modal Shapes of the Steel-Reinforced Concrete Beam

In the NI Example Finder, refer to the Modal Analysis of a Beam VI for more information about performing structural testing with the estimated modal parameters of a time series.

Data Mining

Data mining extracts important features from data and helps you find interesting patterns, rules, or models. Data mining involves a variety of computational methods and techniques. For example, independent component analysis (ICA) is an effective data mining method in the biomedical, mechanical, and seismological fields. You can use ICA to separate informative signals from noise in signals such as electroencephalogram (EEG) signals and magnetoencephalogram (MEG) signals. Refer to the *Understanding Independent Component Analysis* section of Chapter 4, *Performing Statistical Analysis*, for information about performing ICA on a time series.

MEG signals are the magnetic signals generated from electric dipoles around a human brain. Figure 1-6 shows some MEG signals acquired at a human scalp by 148 sensors. These signals indicate brain activities.

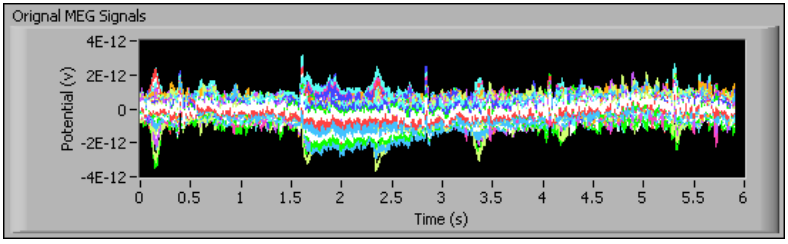

Figure 1-6. MEG Signals from 148 Sensors

All cognitive activities in the human brain generate magnetic signals. Besides those cognitive activities, heartbeats, eye blinking, and breathing also generate magnetic signals. These signals are superimposed on the measured brain signal in Figure 1-6. To distinguish the brain signal from other signals, you can perform ICA on the MEG signals to remove the unwanted signals not originating in the brain activities. Figure 1-7 shows the result of ICA.

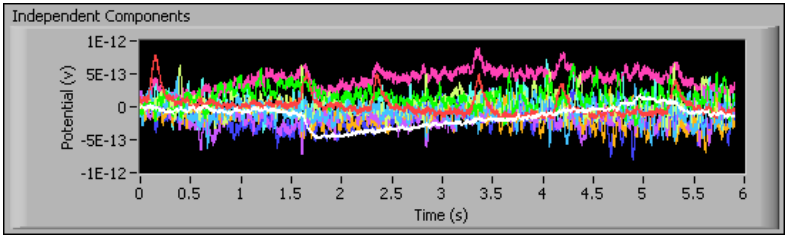

Figure 1-7. The Result of ICA for MEG Signals

You can see that the **Independent Components** graph contains a red line, which clearly indicates the signal generated from heartbeats. You can remove the heartbeat signal from the MEG signals and perform further analysis on the residual signals.

In the NI Example Finder, refer to the MEG Signal Analysis VI for more information about performing ICA to analyze MEG signals.

Industrial Measurement

Industrial measurements involve measuring a variety of physical attributes such as position, speed, and force. In general, you can measure the physical attributes directly with appropriate sensors. However, in some special industrial applications where you cannot apply measuring directly, you have to obtain the physical values using some time series analysis methods, such as correlation.

Figure 1-8 illustrates a speed measurement system for a steel rolling mill. The reflected light from the surface of the steel belt is focused onto two photoelectric cells by lens. The two photoelectric cells, located at different positions with a separation of d, convert the waveform signals of the reflected light into voltage signals. The voltage values from the photoelectric cells form two time series X_t and Y_t. The two series are acquired on the same position track with a separation of d.

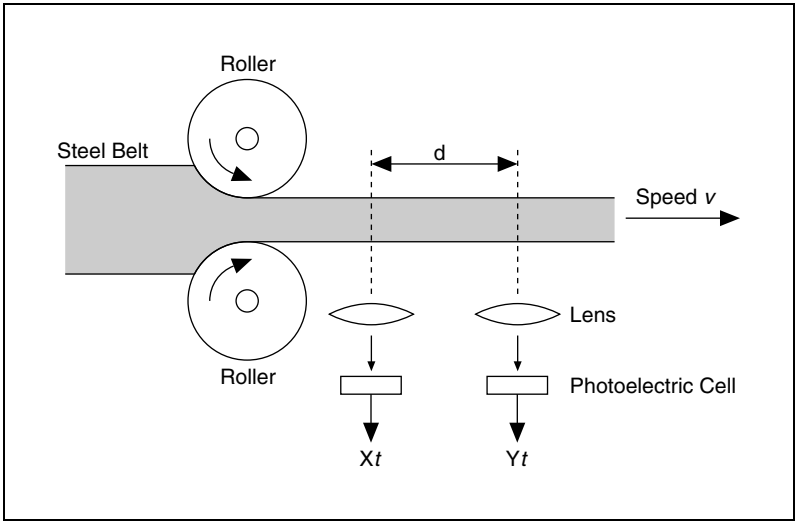

Figure 1-8. Steel Speed Measurement System

To measure the moving speed of the steel belt, you can perform cross-correlation on X_t and Y_t and generate the correlogram, as shown in Figure 1-9. Refer to the *Correlation Methods* section of Chapter 6, *Performing Correlation and Spectral Analysis*, for information about performing cross-correlation on two or more time series.

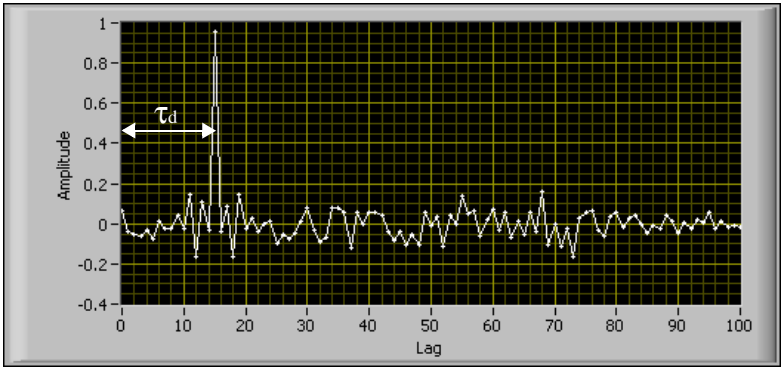

Figure 1-9. Correlogram of Two Time Series from Photoelectric Cells

The correlogram of the two time series contains a maximum point at a lag of τ_d. To compute the speed v of the steel belt, you can use the following equation: $v = d / \tau_d$.

Model Predictive Control

Model predictive control is an important application of time series analysis in engineering. The model predictive control process includes the following steps:

1. Build models of a time series.
2. Use the models to predict the future values of the time series.
3. Make necessary adjustments to the system that generates the time series to make the predicted values align better with target values.

This section provides an example of controlling shaft axes positions based on predicted results.

First, you acquire the positions of the rotating shaft axes to form a time series. You then make a prediction for the next position of the moving shaft by building models of the time series. Using the predicted position, you can take actions to reduce the future position error. Figure 1-10 shows two time series plots of the shaft axes position with prediction control and without

prediction control. The **Shaft Axes Position with Control** graph shows a smaller variance.

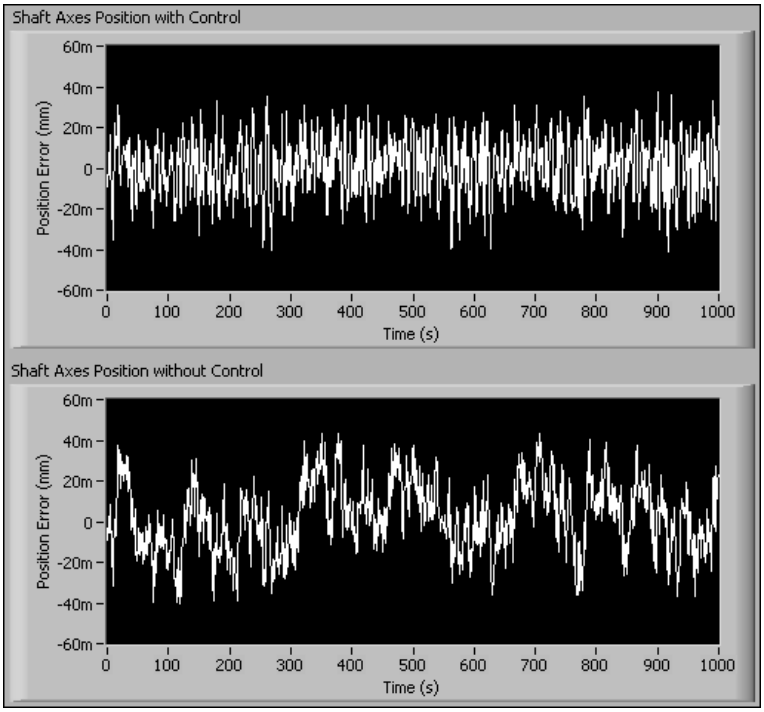

Figure 1-10. Shaft Axes Position Control Based on Time Series Prediction

Overview of LabVIEW Time Series Analysis Tools

The Time Series Analysis Tools provide a collection of VIs that assists you in analyzing scientific and engineering time series and rapidly deploying engineering applications based on the analysis results. You can use these VIs to handle discrete univariate and multivariate (vector) time series.

Time Series Analysis Methods

The Time Series Analysis Tools categorize the time series analysis methods as follows:

- **Statistical analysis methods**—Estimate the statistical parameters of a univariate or multivariate (vector) time series, such as the mean, variance, and skewness values. You can use the estimated statistical parameters to investigate the stochastic characteristics of a physical

system that generates time series. The Time Series Analysis Tools also provide methods for analyzing multivariate time series, such as covariance matrix, independent component analysis (ICA), and principal component analysis (PCA).

Refer to Chapter 4, *Performing Statistical Analysis*, for information about different statistical analysis methods and how to use them.

- **Modeling and prediction methods**—Build dynamic models for a univariate or multivariate (vector) time series and perform predictions based on the estimated models. Building a model helps you understand the characteristics of a physical system that generates time series. Performing a prediction helps you monitor and control the movements of a physical system.

  Refer to Chapter 5, *Building Models and Predicting Time Series Values*, for information about different modeling and prediction methods and how to use them.

- **Correlation and spectral analysis methods**—Detect hidden periodicity or frequency characteristics in a time series or explain the relationship between two time series. The spectral analysis methods help you identify the frequency components in a time series and extract the characteristics of the physical system that generates the time series. You also can estimate the bispectrum, which is useful for detecting nonlinearities in a time series, and the cepstrum, which is useful for deconvolving a time series.

  Refer to Chapter 6, *Performing Correlation and Spectral Analysis*, for information about different correlation and spectral analysis methods and how to use them.

Before applying an analysis method to a time series, you need to preprocess the signal. For example, you need to make sure that the signal contains no low-frequency trends, that the frequency bandwidth is sufficiently narrow, and that the sampling rate is sufficiently high. Use the Preprocessing VIs to preprocess a time series. Refer to Chapter 3, *Acquiring and Preprocessing Time Series*, for information about how to preprocess a time series.

The Time Series Analysis Tools also provide a group of Utilities VIs that you can use to generate time series samples, to scale to an engineering unit, to average the power spectrum, or to load pre-stored data from a file.

Finding Example VIs

The Time Series Analysis Tools provide some example VIs you can use and incorporate into the VIs that you create. You can modify an example VI to fit an application, or you can copy and paste from one or more examples into a VI that you create. You can find the examples using the NI Example Finder. Select **Help»Find Example** to launch the Example Finder. You also can click the arrow on the **Open** button on the **LabVIEW** dialog box and select **Examples** from the shortcut menu to launch the NI Example Finder.

Related Signal Processing Tools

In signal processing, you usually categorize signals into two types: stationary and nonstationary. The spectral content of stationary signals does not change over time, whereas the spectral content of nonstationary signals changes over time. For example, the vibration signal of an engine running at a constant speed is stationary. The vibration signal of an engine at the run-up stage is nonstationary.

Nonstationary signals are categorized into two types according to how the spectral content changes over time: evolutionary and transient. The spectral contents of evolutionary signals change over time slowly. Evolutionary signals usually contain time-varying harmonics. The time-varying harmonics relate to the underlying periodic time-varying characteristic of the system that generates signals. Evolutionary signals also can contain time-varying broadband spectral contents. Transient signals are the short-time events in a nonstationary signal, such as peaks, edges, breakdown points, and start and end of bursts. Transient signals usually vary over time and you typically cannot predict the occurrence exactly.

The LabVIEW Advanced Signal Processing Toolkit contains the following tools and toolkit that you can use to perform signal analysis:

- Time Series Analysis Tools
- LabVIEW Wavelet Analysis Tools
- LabVIEW Time Frequency Analysis Tools
- LabVIEW Digital Filter Design Toolkit

To extract the underlying information of a signal effectively, you need to choose an appropriate analysis tool based on the following suggestions:

- For stationary signals, use the Time Series Analysis Tools or the Digital Filter Design Toolkit. LabVIEW also includes an extensive set of tools for signal processing and analysis. The Digital Filter Toolkit provides tools for designing, analyzing, and simulating floating-point and fixed-point digital filters and tools for generating code for DSP or FPGA targets. Refer to the *Overview of LabVIEW Time Series Analysis Tools* section of this chapter for information about the Time Series Analysis Tools.
- For evolutionary signals, use the Time Frequency Analysis Tools, which include VIs and Express VIs for linear and quadratic time-frequency analysis methods, including the linear discrete Gabor transform and expansion, the linear adaptive transform and expansion, the quadratic Gabor spectrogram, and the quadratic adaptive spectrogram. The Time Frequency Analysis Tools also include VIs to extract features from a signal, such as the mean instantaneous frequency, the mean instantaneous bandwidth, the group delay, and the marginal integration.
- For both evolutionary signals and transient signals, use the Wavelet Analysis Tools, which include VIs and Express VIs for the continuous wavelet transform, the discrete wavelet transform, the undecimated wavelet transform, the integer wavelet transform, and the wavelet packet decomposition. The Wavelet Analysis Tools also include VIs for feature extraction applications, such as denoising, detrending, and detecting peaks and edges.

2

Time Series Analysis Concepts and Methods

This chapter explains the concepts of time series, provides information about how to choose an appropriate method from the LabVIEW Time Series Analysis Tools and describes a typical procedure that you can follow when using the tools to analyze a time series.

Concepts of Time Series

A time series is a sequence of observed values ordered through time. Time series exist in many application areas, ranging from economics to engineering. The Time Series Analysis Tools focus more on the applications in engineering.

Generally, a time series contains the following information:

- The characteristics of the time series, such as amplitude, spectral content and other statistical characteristics.
- The native characteristics or structural parameters of a system that generates the time series, for example, the natural frequency and damping of a civil structure.
- The characteristics of the input or stimulus to the physical system that generates the time series.

Continuous Time Series and Discrete Time Series

In nature, physical quantities such as temperature, pressure, and light intensity change continuously. Observations of these values form a continuous time series.

Given a continuous time series, you can digitize the values at a specified time interval to obtain a discrete time series. Figure 2-1 shows the seismograph of the Kobe earthquake, recorded at Tasmania University, Hobart, Australia on January 16, 1995. In this figure, the continuous earthquake vibration signal is sampled at a one-second interval to form a discrete time series.

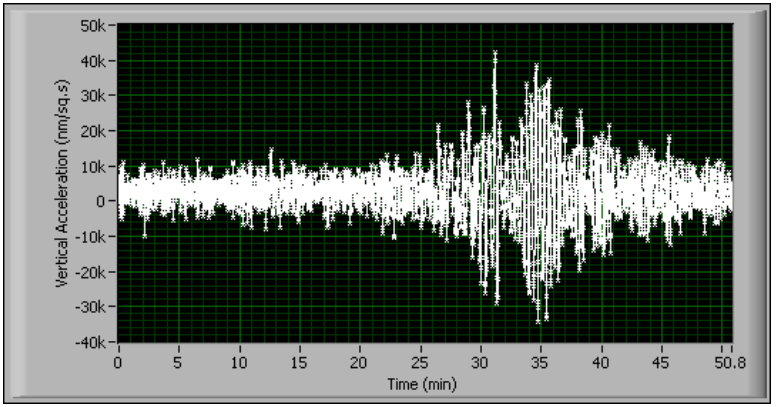

Figure 2-1. Seismograph of the Kobe Earthquake[1]

Time-Ordered Series and Spatial-Ordered Series

Time series can be ordered not only through time but also through other physical units. For example, you can obtain a discrete time series ordered versus angular position by sampling the diameter of a spindle as a function of angle.

Figure 2-2 shows an example of the diameter error as a function of angle of a spindle during a lathe machining process. The diameter error generates a discrete time series ordered versus angular position.

[1] Data source: Hyndman, R.J. (n.d.) *Time Series Data Library*, `http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/`

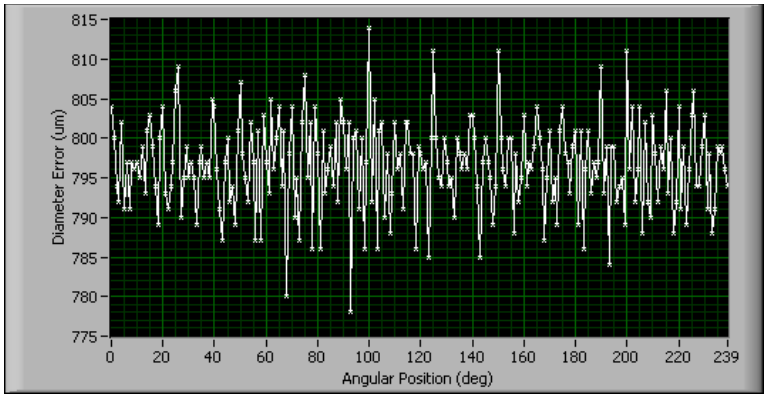

Figure 2-2. Diameter Error of a Spindle

Univariate Time Series and Multivariate Time Series

You can collect observed values from a single source or simultaneously from two or more sources. Single-source observations generate univariate time series, and multi-source observations form multivariate time series, or vector time series. For example, you can obtain a multivariate time series by recording the values of pressure, flow, and temperature simultaneously in an industrial process.

Figure 2-3 shows an example of the vibration signals from a steel-reinforced concrete beam. The signals are acquired simultaneously from seven acceleration sensors located at different positions on the beam. Refer to the *Structural Testing* section of Chapter 1, *Introduction to Time Series Analysis*, for information about how to acquire these vibration signals with sensors. The simultaneously-sampled vibration signals form a multivariate discrete time series.

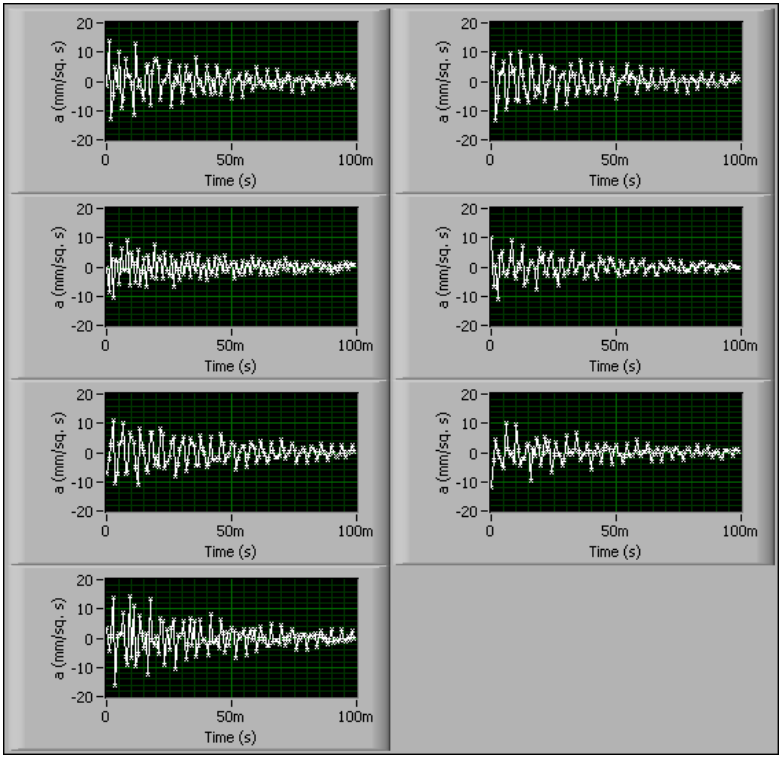

Figure 2-3. Vibration Signals from a Steel-Reinforced Concrete Beam

Stationary Time Series and Nonstationary Time Series

In theory, given a behavioral model for a system, you can predict future values of a time series measured from that system, based on past observations. However, in practice, physical systems are affected by many kinds of disturbances, so the predicted values always reflect the stochastic, or statistical, characteristic of a time series.

Generally speaking, if the statistical characteristic of a time series contains no systematic change, the time series is stationary. Otherwise the time series is nonstationary.

Choosing an Appropriate Method

The Time Series Analysis Tools contain a collection of Time Series Analysis VIs with different methods. Refer to the *Time Series Analysis Methods* section of Chapter 1, *Introduction to Time Series Analysis*, for information about available time series analysis methods.

Each of these methods is classified as either time domain or frequency domain. You can select appropriate methods from these two classes according to the analysis objective.

All of the statistical analysis methods introduced in Chapter 4, *Performing Statistical Analysis*, the modeling and prediction methods introduced in Chapter 5, *Building Models and Predicting Time Series Values*, and the correlation methods introduced in Chapter 6, *Performing Correlation and Spectral Analysis*, are time-domain methods. You can use the statistical analysis methods to investigate the stochastic characteristics of a time series. Stochastic characteristics, for example, are helpful in quality controls in manufacturing production. If you have two or more related time series, you can analyze them jointly using a covariance matrix, principal component analysis (PCA), or cross-correlation method to investigate their relatedness. Independent component analysis (ICA) can separate independent signals from linearly mixed data. The modeling methods help you build parametric behavioral models for time series, which help you predict or control future values.

The spectral analysis methods introduced in Chapter 6, *Performing Correlation and Spectral Analysis*, are frequency-domain methods. You can use the nonparametric or model-based spectral analysis methods to investigate the vibration characteristics of physical systems, such as resonance frequencies and harmonic frequencies. Some of the methods also support multivariate time series, such as the MUSIC method, which computes the common spectral components existing in a multivariate vibration time series.

Time Series Analysis Procedure

Figure 2-4 illustrates the procedure that you can follow when using the Time Series Analysis Tools to analyze a time series.

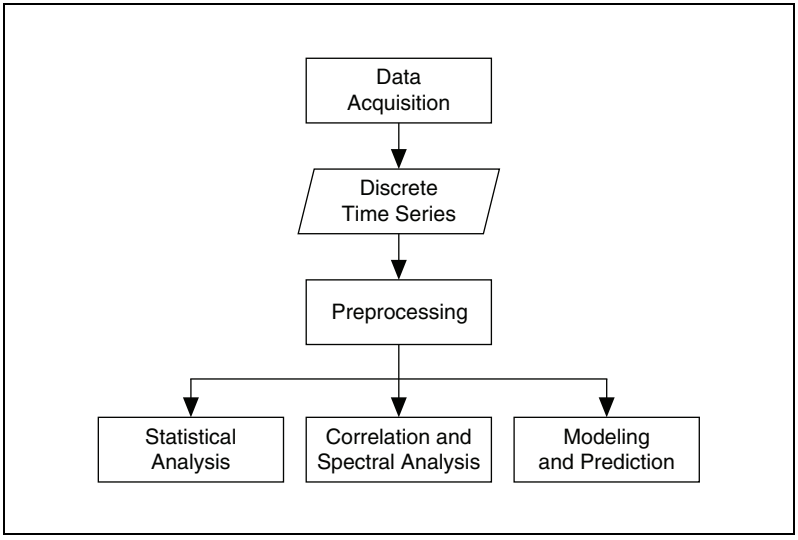

Figure 2-4. General Procedure of Time Series Analysis

A typical time series analysis procedure includes the following steps:

1. Acquire a discrete time series through NI-DAQ or by loading existing data from a file. Refer to Chapter 3, *Acquiring and Preprocessing Time Series*, for information about how to acquire a discrete time series.
2. Preprocess the time series if necessary; for example, you can resample the time series using a different time interval, or remove a low-frequency trend from the time series. Refer to Chapter 3, *Acquiring and Preprocessing Time Series*, for information about the methods that you can use to preprocess a time series.
3. Obtain useful information from the preprocessed time series by selecting suitable time series analysis methods from the following categories:
   - Statistical analysis
   - Correlation and spectral analysis
   - Modeling and prediction

   Refer to the *Choosing an Appropriate Method* section of this chapter for information about how to select a suitable time series analysis method.

3

Acquiring and Preprocessing Time Series

The first step in time series analysis is to obtain the time series that you want to analyze. You can acquire data from National Instruments data acquisition hardware and software or from files that contain pre-stored data. After you obtain a time series, you need to perform some preliminary processing to make the raw data suitable for further analysis. The LabVIEW Time Series Analysis Tools provide VIs that you can use to perform data inspection and preprocessing.

This chapter describes how to obtain discrete time series and how to perform time series inspection and preprocessing with the Time Series Analysis Tools.

Obtaining a Discrete Time Series

Signals from physical systems are typically continuous. These real-world signals, such as earthquake waveforms in earthquake monitoring, vibration signals from mechanical devices, or electroencephalogram (EEG) signals, are sampled to form discrete-time representations to enable computer processing. You can sample a continuous time series to form a corresponding discrete time series using data acquisition hardware, such as NI-DAQ devices.

If you have pre-stored data files, use the TSA Read from Files VI to load 1D or 2D arrays of time series data into LabVIEW. The TSA Read from Files VI supports spreadsheet text files and WAV files. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about how to use this VI.

Note Use the Read LabVIEW Measurement File Express VI to read data from `.lvm` files. Use the Read Waveforms from File VI to read data if you used the Write Waveforms to File VI to save the data. Refer to the *LabVIEW User Manual* for information about file I/O.

The following section describes the factors that influence the quality of a discrete time series.

Factors that Influence a Discrete Time Series

To ensure that the obtained discrete time series accurately represents the information contained in the original continuous time series, you need to consider the following factors when sampling data from a continuous time series:

- **Sampling rate**—According to the Nyquist sampling theorem, if you want to retain the information that a continuous signal contains, the sampling rate must be greater than twice the highest frequency component in the original signal to avoid frequency aliasing. The Nyquist frequency is half of the sampling rate of the discrete time series.

  If you acquire a signal with an NI Dynamic Signal Acquisition (DSA) device, you get automatic aliasing protection. The signal first passes through fixed analog filters that remove the frequency components beyond the range of the analog-to-digital (A/D) converter; then digital anti-aliasing filters automatically adjust the cutoff frequency to remove any frequency component above the Nyquist frequency.

  If you acquire a signal with other NI DAQ hardware, you can avoid aliasing with any of the following methods:

  – Increase the sampling rate until the Nyquist frequency exceeds the highest frequency component in the signal.

  – Apply an external lowpass filter with NI SCXI models.

  – Use an inherently bandlimited sensor.

  Refer to the *LabVIEW Measurements Manual* for information about signal measurement and data acquisition.

- **Number of samples**—The number of samples, or the length of a time series, limits how fine the frequency resolution can be. A time series with a large number of samples can provide fine frequency resolution. You usually specify a necessary number of samples based on the following formula:

$$N \geq \frac{f_s}{\Delta f}$$

  where N is the number of samples, f_s is the sampling rate, and Δf is the required minimum frequency resolution.

- **Amplitude resolution**—The amplitude of a discrete time series for computer processing also is discrete because you usually acquire the time series with an A/D converter. Quantization error of the A/D converter is a correlated and nonlinear impairment that reduces measurement quality. You must make certain that the data has sufficiently fine amplitude resolution. Refer to the *LabVIEW Measurements Manual* for information about quantization error.

Preprocessing a Discrete Time Series

Preprocessing helps you make an acquired discrete time series more suitable for further analysis. The Time Series Analysis Tools provide the Preprocessing VIs that enable you to smooth a time series, to resample a time series, or to remove the trend from a time series. The Preprocessing VIs include the Time Series Preprocessing Express VI that you can use to select an appropriate method to preprocess a time series interactively. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about these VIs.

Resampling a Time Series

When you acquire a discrete time series, to avoid frequency aliasing, the sampling rate must be greater than twice the highest frequency component of the source signal. If you want to build models for a time series, you usually specify a sampling rate ten times as large as the highest frequency component of the source signal when acquiring the time series. However, a much higher sampling rate substantially increases the computation burden. If the sampling rate is unnecessarily high, you can resample the acquired time series and generate a new time series with a lower sampling rate.

Sometimes the time series under analysis is unequally-sampled. To use time series analysis methods, you need to resample the time series at equal time intervals to generate an equally-sampled time series.

Use the TSA Resampling VI to resample a time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Avoiding Frequency Aliasing

Before resampling, the frequency bandwidth of the source signal must be less than the Nyquist frequency at the new sampling rate to avoid aliasing. If the time series contains frequency components whose frequency bands are greater than the new Nyquist frequency, you can use a lowpass filter to attenuate those frequency components that are greater than the new Nyquist frequency.

Figure 3-1 shows a time series that contains a frequency component from 100 to 200 Hz and another frequency component from 300 to 400 Hz. The sampling rate of the time series is 1000 Hz.

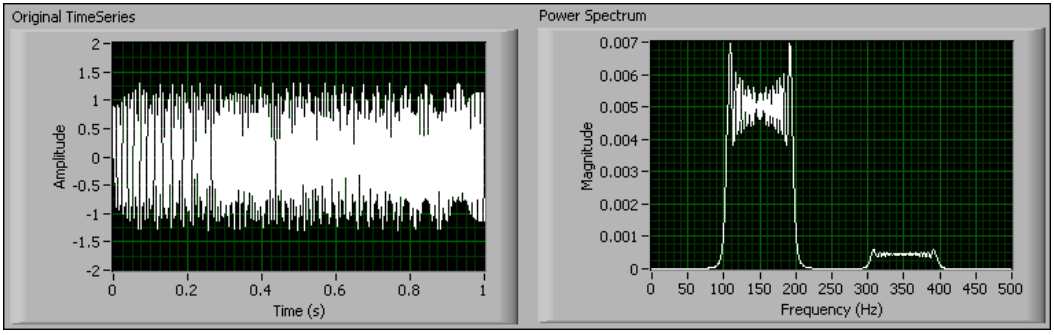

Figure 3-1. Original Time Series and the Power Spectrum

If the frequency band of interest is from 0 to 250 Hz, you can reduce the sampling rate to 500 Hz. When you resample the time series using the new sampling rate, frequency aliasing occurs if you do not attenuate the frequency component from 300 to 400 Hz because this frequency component is above 250 Hz, the new Nyquist frequency.

Figure 3-2 shows the resampled time series that was not properly filtered before resampling and therefore contains frequency aliasing. In the **Power Spectrum** graph, you can see that frequency aliasing distorts the original frequency component from 100Hz to 200Hz.

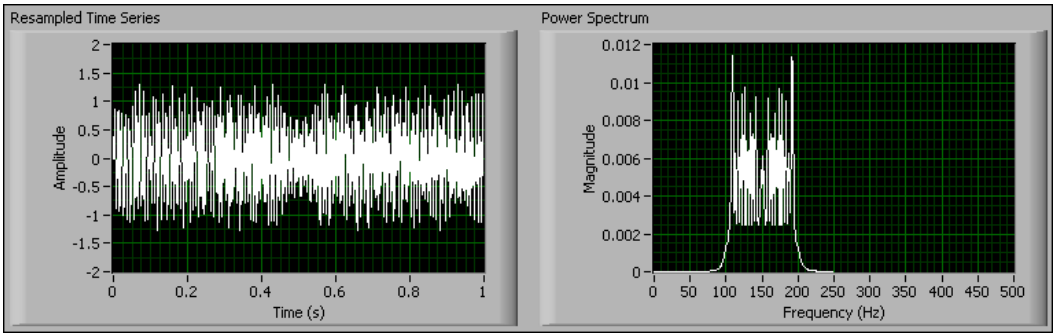

Figure 3-2. Resampled Time Series and the Power Spectrum

To avoid frequency aliasing in the resampling operation, you first must sufficiently attenuate the frequency component that is above the new Nyquist frequency. In this example, you need to use a lowpass filter to attenuate the frequency component from 300 to 400 Hz in the original time series. Refer to the *LabVIEW Analysis Concepts* manual for information about filtering data.

Figure 3-3 shows the filtered time series and the power spectrum. Notice that the lowpass filter removes the frequency component from 300 to 400 Hz from the time series.

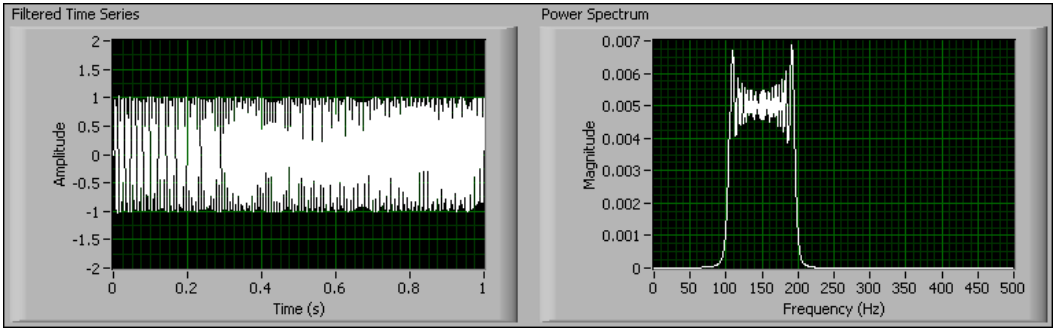

Figure 3-3. Filtered Time Series and the Power Spectrum

After removing the frequency component that is above the new Nyquist frequency, you can resample the time series with the new sampling rate of 500 Hz without frequency aliasing. The **Power Spectrum** graph of Figure 3-4 shows that the resampled time series preserves the frequency components of interest from 0 Hz to 250 Hz without distortion.

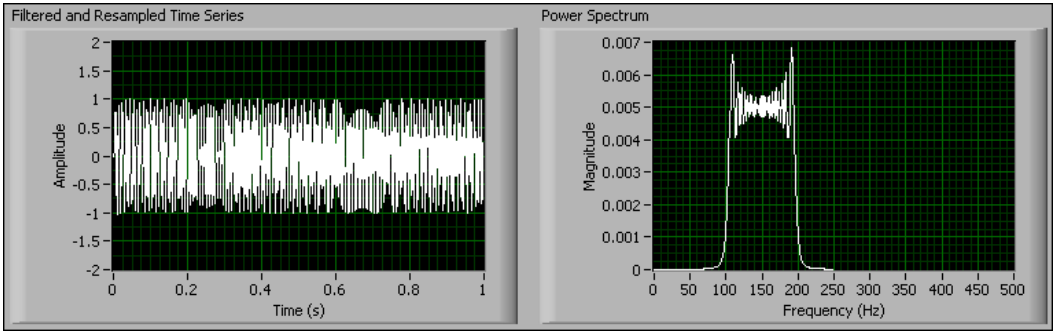

Figure 3-4. Filtered and Resampled Time Series and the Power Spectrum

Converting an Unequally-Sampled Time Series

Time series analysis methods process only equally-sampled time series. To analyze an unequally-sampled time series, you need to convert the unequally-sampled time series into an equally-sampled time series using the TSA Resampling VI.

Figure 3-5 shows an unequally-sampled time series and the corresponding equally-sampled time series. You can see that the time indexes are distributed equally in the **Resampled Time Series** graph.

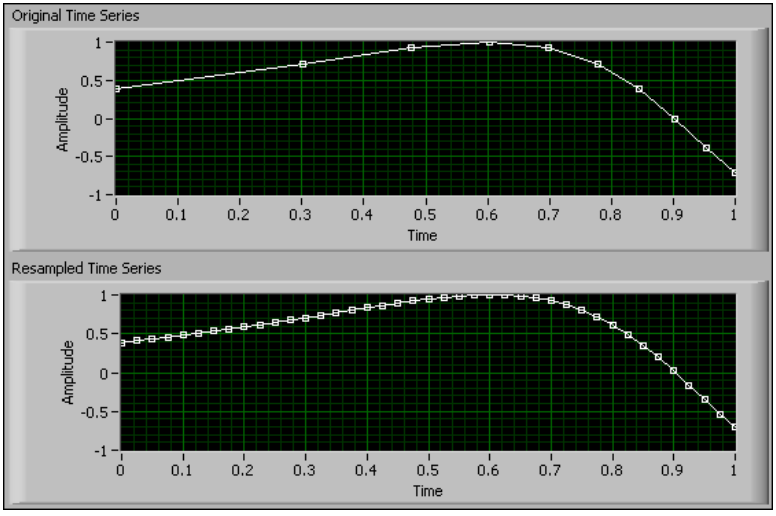

Figure 3-5. Unequally-Sampled and Equally-Resampled Time Series

In the **Browse** tab of the NI Example Finder, you can view this example by selecting **Toolkits and Modules»Time Series Analysis» Getting Started»Resampling Unequally-Sampled Time Series VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Time Series Analysis*, for information about launching the NI Example Finder.

Smoothing a Time Series

Using the Time Series Analysis Tools, you can smooth a time series with either the moving average method or the exponential average method.

The moving average method estimates the local averaged value based on the adjacent values with a finite impulse response (FIR) filter. You can use this method to remove the noise disturbance from a time series.

Use the TSA Moving Average VI to perform a moving average. This VI provides two typical moving average filters—Spencer and Henderson. You also can customize the coefficients of the moving average filters. The TSA Moving Average VI compensates the phase shift of the smoothed time series so no phase delay exists between the original and the smoothed time series. Refer to the *LabVIEW Help*, available by selecting **Help» VI, Function, & How-To Help**, for information about this VI.

Exponential averaging is another common approach to producing a smooth time series, which helps you remove the variations that the original time series contains. Exponential averaging also can remove seasonality, which is low-frequency periodic spectral content in a time series.

Use the TSA Exponential Average VI to perform exponential smoothing operations on a time series. You can select a suitable smoothing scheme according to the characteristics of the time series. This VI provides the following exponential smoothing schemes:

- Single exponential smoothing scheme—Suitable for a time series that does not contain a systematic trend or seasonality.
- Double exponential smoothing scheme—Suitable for a time series that contains a systematic trend but does not contain seasonality.
- Triple exponential smoothing scheme—Suitable for a time series that contains both a systematic trend and seasonality.

Figure 3-6 shows the results of exponential smoothing with different schemes. This figure indicates that the triple scheme follows the time series much closer than the single and double schemes because the time series contains a systematic trend and seasonality.

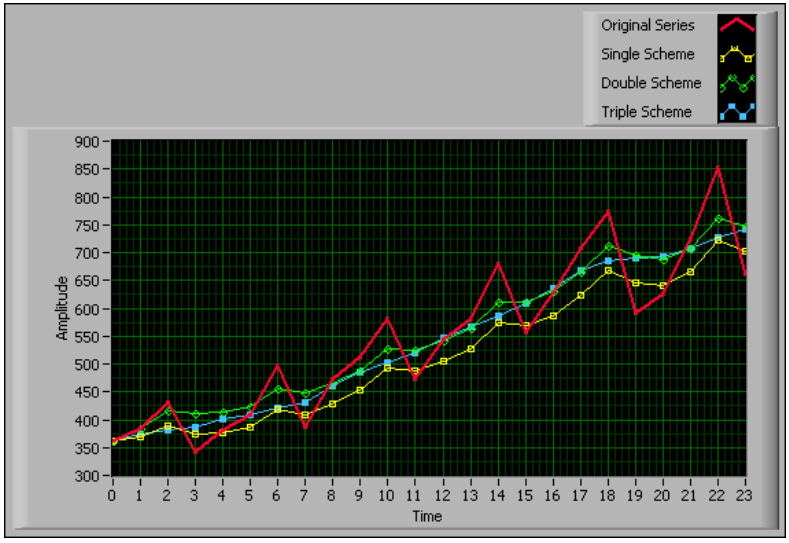

Figure 3-6. The Result of Time Series Exponential Smoothing

When using the triple exponential smoothing scheme, you need to specify the season type of the analyzed time series. Figure 3-7 shows two time series with different types of seasonality—additive and multiplicative.

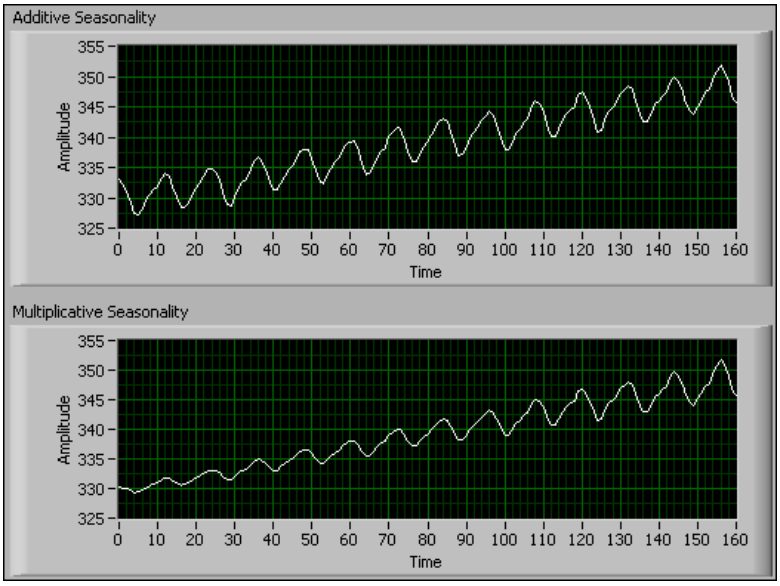

Figure 3-7. Time Series with Additive or Multiplicative Seasonality

In Figure 3-7, the **Additive Seasonality** graph shows a time series that has a constant amplitude change in seasonality. Using the TSA Exponential Average VI, you can analyze this type of time series by specifying **Additive** in **season type**. The **Multiplicative Seasonality** graph shows a time series that has a seasonality with the amplitude increasing over time. You can analyze this type of time series by specifying **Multiplicative** in **season type**.

Detrending a Time Series

A time series usually contains some constant amplitude offset components or low-frequency trends. The constant-offset components and low-frequency trends do not affect the dynamic characteristics of the system being analyzed, and the amplitudes of these trends sometimes are large and corrupt the results of time series modeling. Therefore, you need to remove the constant-offset components or low-frequency trends before performing further analysis.

If a time series contains no long-term (low-frequency) trends but only constant-offset components, you can detrend this time series by subtracting the mean value.

If a time series contains long-term trends and constant-offset components, use the TSA Detrend VI to obtain a detrended time series. This VI estimates the trend of a time series with the curve-fitting methods. Refer to the *LabVIEW Analysis Concepts* manual for information about curve fitting.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about the TSA Detrend VI.

4

Performing Statistical Analysis

Statistical analysis methods enable you to investigate the stochastic characteristics of a time series. You can use the computed statistical values to infer some underlying information about the associated physical systems, for example, if a machining tolerance error is due to a systematic anomaly or random fluctuations. The computed statistical values for a time series also help you judge if other analysis techniques, such as modeling, are useful in analyzing that time series.

This chapter describes the statistical analysis methods that the LabVIEW Time Series Analysis Tools provide. You can process both univariate and multivariate time series with these methods.

Basic Statistical Analysis Methods

The basic statistical analysis methods enable you to compute statistical values for a univariate or multivariate time series, including mean, variance, skewness, kurtosis, confidence limits, stationarity, and whiteness.

Note LabVIEW Full and Professional Development Systems also contain Probability and Statistics VIs that implement similar statistical analysis methods. The Probability and Statistics VIs support univariate signals only. The Statistical Analysis VIs in the Time Series Analysis Tools provide more functionality and more options, such as geometric average and trimmed average, than the Probability and Statistics VIs in LabVIEW.

Understanding the Mean and Variance Values

Mean and variance are fundamental statistical attributes of a time series. The arithmetic mean of a time series is the average or expected value of that time series. In some cases, the mean value of a time series can be the operating point or working point of a physical system that generates the time series. In general, you must subtract the value of the operating point from a time series before building predictive behavioral models for that time series. Median and trimmed mean are more robust than arithmetic mean and geometric mean if the raw data does not contain significant outliers. Use the TSA Mean VI to obtain the median or different mean values of a time series.

The variance of a time series measures the dispersion of the time series data samples around the mean value. Use the TSA Deviation and Variance VI to obtain the variance value of a time series.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about the TSA Mean VI and the TSA Deviation and Variance VI.

Understanding the Skewness and Kurtosis Values

The skewness and kurtosis are higher-order statistical attributes of a time series. Skewness indicates the symmetry of the probability density function (PDF) of the amplitude of a time series. A time series with an equal number of large and small amplitude values has a skewness of zero. A time series with many small values and few large values is positively skewed (right tail), and the skewness value is positive. A time series with many large values and few small values is negatively skewed (left tail), and the skewness value is negative.

Figure 4-1 illustrates two time-series PDFs with opposite skewness. Each bin in these histograms indicates the probability that any particular sample from the corresponding time series has a value within the range of that bin.

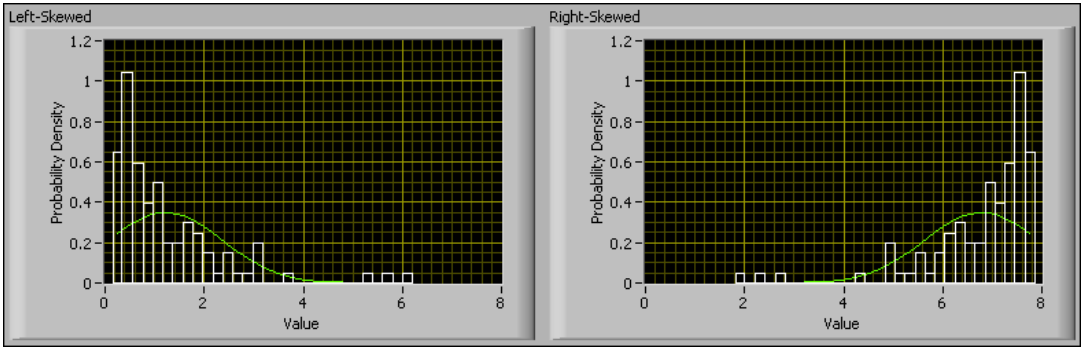

Figure 4-1. Time Series with Left-Skewed and Right-Skewed Distributions

The **Left-Skewed** graph shows a time series that is positively skewed. The **Right-Skewed** graph shows a time series that is negatively skewed. The green lines in these plots are the best-fit normal distributions for the given distributions.

Kurtosis measures the peakedness of the PDF of a time series. A kurtosis value close to three indicates a Gaussian-like peakedness. PDFs with relatively sharp peaks have kurtosis greater than three. PDFs with relatively

flat peaks have kurtosis less than three. Figure 4-2 illustrates two time series with different peakedness.

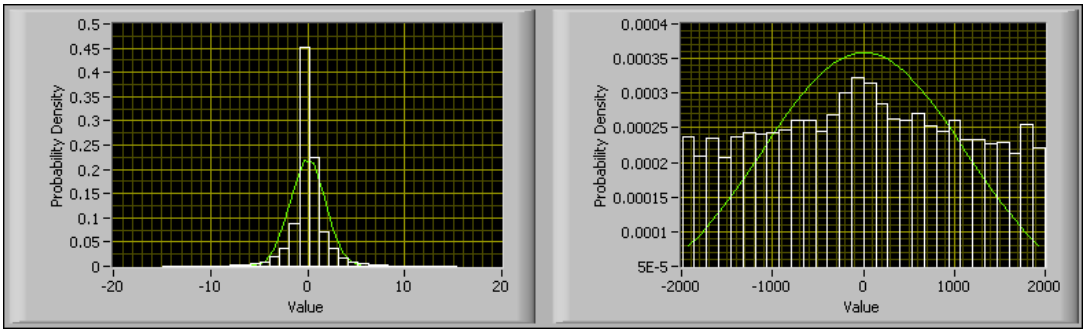

Figure 4-2. Time Series with Peaked and Flat Distributions

The histogram on the left shows a time series with a peaked distribution (kurtosis = 10.9817). The histogram on the right shows a time series with a flat distribution (kurtosis = 1.90294). The green lines in these plots are the Gaussian PDFs with the kurtosis equal to three.

Changes in skewness and kurtosis of a time series indicate systematic changes in the associated physical systems. Use the TSA Skewness and Kurtosis VI to obtain the skewness and kurtosis values of a time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Understanding the Confidence Limits

Confidence limits are user-specified probability bounds for the range of statistical parameters. Confidence limits are useful because the estimated values of statistical parameters, such as mean and variance, typically differ for different sections of a time series. Instead of calculating one value for mean and standard deviation for an entire time series, you can generate a range of values for the mean and standard deviation over that time series by specifying confidence limits.

The estimation interval for a given set of confidence limits indicates how much the calculated mean and standard deviation values can vary across the time series. A narrow estimation interval for a parameter implies that the parameter varies little across the time series.

Use the TSA Confidence Limits VI to perform interval estimations on the statistical parameters of a time series. Refer to the *LabVIEW Help*,

available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Testing the Normal Distribution

A time series whose PDF approximates a Gaussian distribution are called normally distributed. Many statistical analysis methods work well only for time series that are normally distributed. Furthermore, some modeling methods assume that the measurement noise in a time series is Gaussian white noise. For these methods, you can measure how well the histogram of a time series or measurement noise approximates a normal distribution and determine appropriate statistical analysis and modeling methods.

A normal distribution test plots the calculated distribution for a time series and its best-fit normal distribution together. If the two distributions show a close correlation, the calculated distribution for the time series is normally distributed.

Use the TSA Normal Distribution Test VI to test how normally a time series is distributed. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Checking the Stationarity and Whiteness

Time series analysis methods can produce satisfactory analysis results for stationary time series but might not provide satisfactory analysis results for nonstationary signals. You can run a stationarity test to check if a time series is stationary.

For nonstationary signals, use other analysis methods such as joint time-frequency analysis (JTFA) and wavelet analysis. Refer to the *Time Frequency Analysis Tools User Manual* or the *Wavelet Analysis Tools User Manual* for information about analyzing nonstationary signals.

Tip You potentially can transform a nonstationary time series into a stationary time series by removing any trend or slow variation from the time series. You also can slice the time series into approximately stationary sections and separately analyze each section.

Whiteness tests examine the spectral flatness of a time series and indirectly indicate statistical independence of a time series data samples. You also can use whiteness tests to analyze the fitness of a particular time-series model created with some modeling methods. For those methods, if the residual estimation noise for a particular model is white with a specified confidence level, the estimated model is a good fit.

Use the TSA Stationarity Test VI and the TSA Whiteness Test VI to check the stationarity and whiteness of a time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about these VIs.

Multivariate Statistical Analysis Methods

Multivariate statistical analysis methods enable you to investigate statistical interdependence between variables in a multivariate time series. These methods also help you perform blind source separation or eliminate redundant or extraneous variables in a multivariate time series. You also can use these methods to transform a multivariate time series so that information can be concentrated in a smaller number of variables, which enable you to reduce the dimensionality of multivariate time series.

Understanding the Covariance Matrix

Covariance matrices measure the correlation between two or more time series acquired during the same period. In a unified covariance matrix or correlation-coefficient matrix, the diagonal values all have a value of one because all signals are correlated with themselves perfectly. Nondiagonal values close to one indicate that the corresponding variables are highly correlated.

Use the TSA Covariance VI to compute the covariance matrix and unified covariance matrix for multivariate time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Understanding Principal Component Analysis

The main purpose of principal components analysis (PCA) is to enable you to isolate and remove extraneous or redundant variables in a multivariate time series. Extraneous and redundant variables increase the dimensionality of a time series and prevent you from finding important underlying patterns in the data. With the PCA method, you can reduce the dimensionality of a time series and retain as much information as possible. You also can make underlying patterns in the data more explicit and easier to find. PCA is useful in applications such as pattern recognition and image compression.

Figure 4-3 shows a simulated multivariate time series that contains two variables. These two variables are uncorrelated with each other.

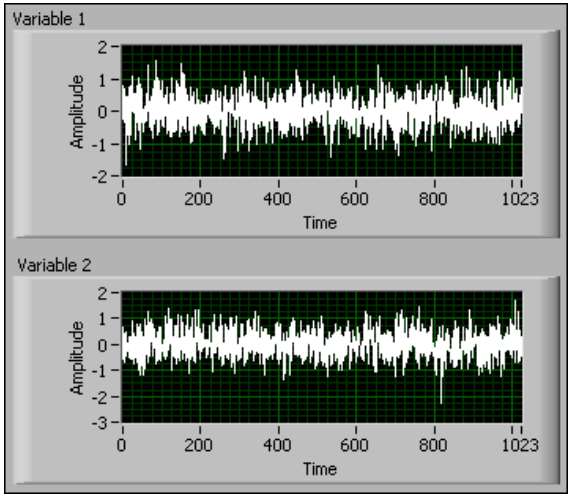

Figure 4-3. Uncorrelated Variables in a Multivariate Time Series

Figure 4-4 shows the correlation and unified covariance matrix of the two variables. The **Correlation** graph is an XY graph that uses **Variable 1** as x-axis and **Variable 2** as y-axis. **Variable 1** and **Variable 2** are uncorrelated with each other because the data points spread across the XY graph irregularly. The nondiagonal elements in the unified **Covariance Matrix** are close to zero, so the two variables are uncorrelated.

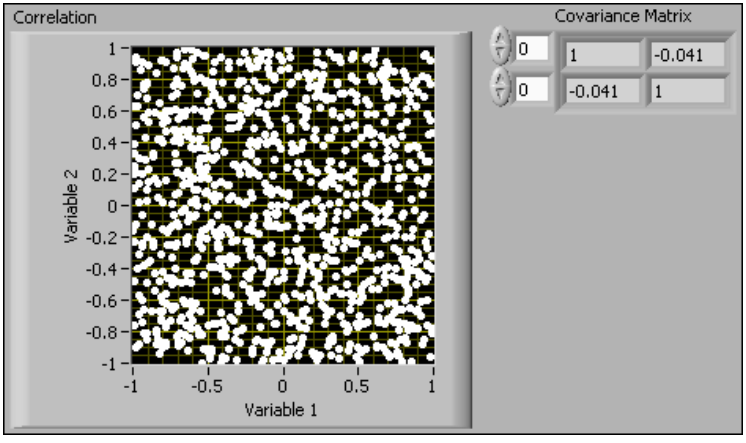

Figure 4-4. Correlation Graph and Covariance Matrix of Uncorrelated Variables

Figure 4-5 shows a simulated multivariate time series that contains two variables. These two variables are correlated with each other.

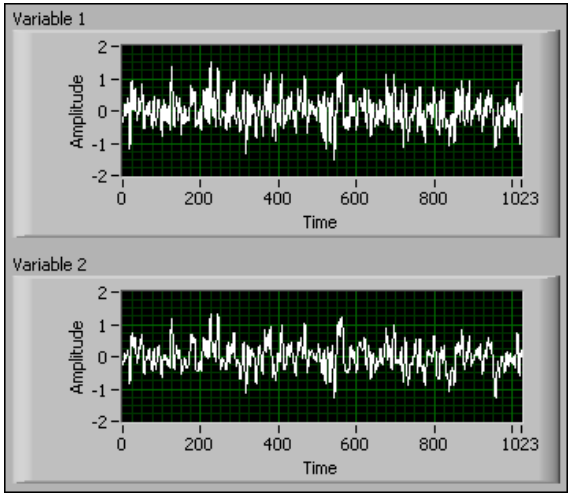

Figure 4-5. Correlated Variables in a Multivariate Time Series

Figure 4-6 shows the correlation and unified covariance matrix of the two variables. In the **Correlation** graph, **Variable 1** and **Variable 2** increase together. The nondiagonal elements in the unified **Covariance Matrix** are close to one, so the two variables are highly correlated.

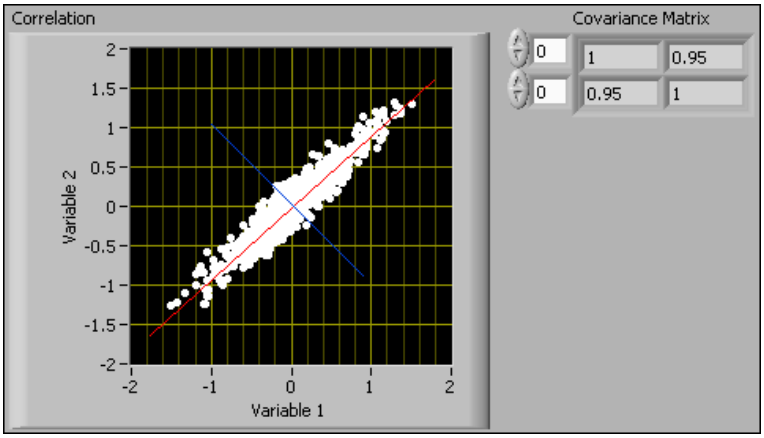

Figure 4-6. Correlation Graph and Covariance Matrix of Correlated Variables

PCA transforms correlated time series into uncorrelated time series. In the case of a two-variable time series, the first principal component is the line along the direction that has maximum variance. The second principal component is the line along the direction that has second largest variance and is perpendicular to the first principal component. In Figure 4-6, the red and blue lines represent the first principal component and the second principal component, respectively. The two lines form a new coordinate system. PCA rotates the original coordinate system to the new coordinate system and reduces the correlation between variables, potentially enabling you to eliminate the component with the lower variance if you judge the variance to be negligible in the application.

Figure 4-7 shows the correlation plot and unified covariance matrix of the resulting time series from PCA.

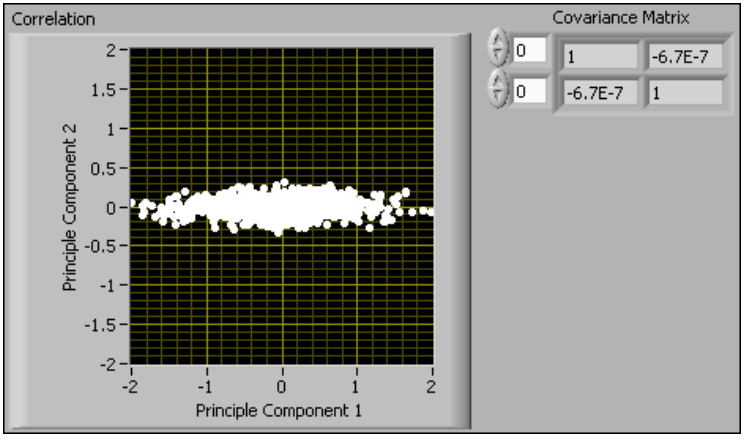

Figure 4-7. Correlation Plot and Covariance Matrix after PCA

PCA is not limited to two-variable time series. When the number of variables is greater than two, the eigenvectors of the correlation matrix are principal components. PCA reorders the eigenvectors based on the corresponding eigenvalues and projects the original time series on the eigenvectors. Principal component scores describe the projection results of the time series on the new coordinate system. Principal component scores are the linear combination of variables in the original time series.

The following is an example that uses PCA for image compression. Figure 4-8 shows an x-ray image. You can consider each row of the image as a time series and the whole image as a multivariate time series.

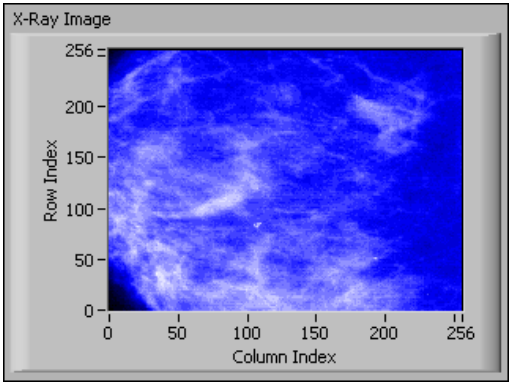

Figure 4-8. X-Ray Image

You can check the correlation between each row of the image by computing the unified covariance matrix. Figure 4-9 shows part of the unified covariance matrix for the image in Figure 4-8. The nondiagonal elements are close to one. Therefore, strong correlation exists among some of the rows of the image. You can reduce the correlation by performing PCA on the original image.

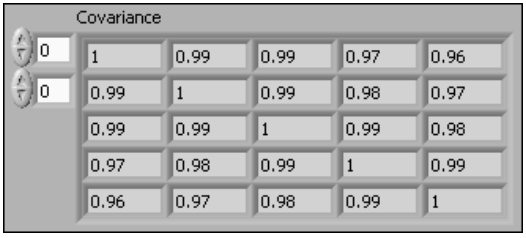

Figure 4-9. Unified Covariance Matrix of X-Ray Image

Figure 4-10 shows the PCA result of the image. The variance of each row decreases as the row index increase. The rows contain less significance as the row index increases. Deciding which rows are significant or useful is a matter of judgment, but a reasonable threshold is at row 50. If you keep only the first 50 rows, you can reduce the data dimensionality and corresponding image size in bits while retaining important information in the image.

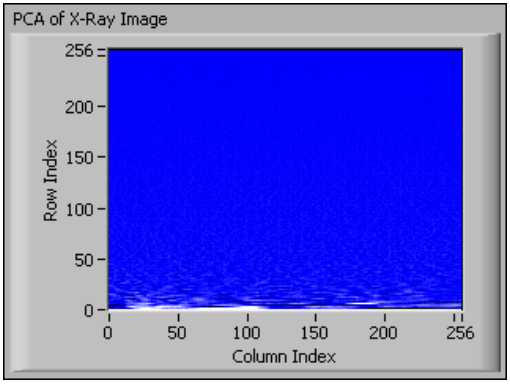

Figure 4-10. PCA Result of X-Ray Image

Figure 4-11 shows the reconstructed image with the principal component scores of only the first 50 principal components. You can see that the reconstructed image properly retains the major features of the original image.

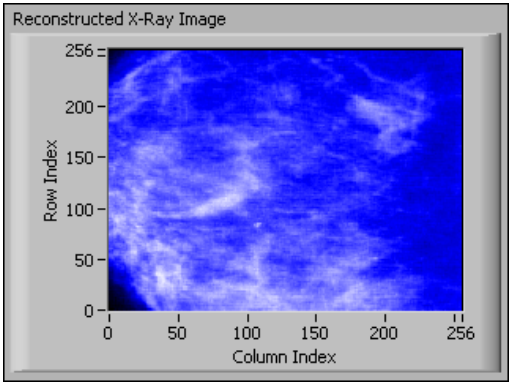

Figure 4-11. Reconstructed X-Ray Image from First 50 Principal Components Scores

Use the TSA Principal Component Analysis VI to perform PCA on multivariate time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

In the **Browse** tab of the NI Example Finder, you can view this example by selecting **Toolkits and Modules»Time Series Analysis»Applications» Image Compression with PCA VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Time Series Analysis*, for information about launching the NI Example Finder.

Note The PCA technique is a linear transform. You cannot use PCA to process a multivariate time series that contains too much nonlinear correlation between variables.

Understanding Independent Component Analysis

Independent component analysis (ICA) generates a multivariate time series with statistically-independent components from an original multivariate time series with statistically-dependent components. ICA is a generalization of PCA. ICA removes not only the second-order statistical dependency but also high-order statistical dependencies between the variables of a multivariate time series. However, PCA removes only the second-order statistical dependency between the variables.

One typical application of ICA is blind source separation, or revealing independent sources from sensor observations that are unknown linear mixtures of the unobserved source signals. Figure 4-12 illustrates the flowchart of blind source separation.

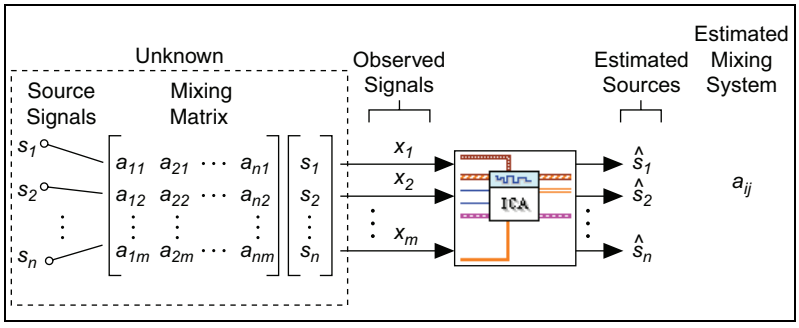

Figure 4-12. Flowchart of Blind Source Separation

In Figure 4-12, the observed signals are the linear mixtures of a set of unknown independent source signals. ICA estimates the source signals and the mixing matrix with only the observed signals. The estimated source signals are called independent components because they are statistically independent of each other.

For example, the electroencephalogram (EEG) data are recordings of electrical potentials at many different locations on a human scalp. These electrical potentials are the mixtures of signals generated by brain activities. ICA can help you recover the components of brain activities and reveal underlying information about those activities.

Another application of ICA is removing artifacts from signals. For example, in biomedical signal analysis, the magnetoencephalography (MEG) signals from a human brain usually contain artifacts such as eye movements, heartbeats, and measurement noise. You can use ICA to remove the artifacts and enhance the MEG signals. Using ICA to remove artifacts usually involves the following steps:

1. Compute the separating matrix, that is, the inverse of the mixing matrix, and obtain the independent components.
2. Remove undesirable independent components by setting their values to zeros.
3. Reconstruct signals from independent components with the mixing matrix.

You also can use ICA in the applications of feature extraction and data mining because ICA can make the features more explicit in the resulting independent components.

Use the TSA Independent Component Analysis VI to perform ICA on multivariate time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

In the **Browse** tab of the NI Example Finder, you can view the example of processing MEG signals by selecting **Toolkits and Modules» Time Series Analysis»Getting Started»Independent Component Analysis VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Time Series Analysis*, for information about launching the NI Example Finder.

5

Building Models and Predicting Time Series Values

One of the most important applications of time series analysis is building mathematical models for observed time series. The resulting mathematical models can help you better understand the dynamic characteristics of the corresponding physical systems and assist you in monitoring or providing feedback control for the systems. You also can use the resulting mathematical models to estimate the power spectrum of a time series. Refer to Chapter 6, *Performing Correlation and Spectral Analysis*, for information about model-based spectral analysis methods.

This chapter introduces various models you can build for univariate or multivariate time series using the LabVIEW Time Series Analysis Tools. This chapter also discusses the theoretical background of estimating models and describes the relationship between model coefficients and dynamic characteristics of a time series.

Building Models

Using the Time Series Analysis Tools, you can build the following types of models:

- **Polynomial models**—For univariate time series, you can build autoregressive (AR) models, moving average (MA) models, and autoregressive-moving average (ARMA) models. For multivariate (vector) time series, you can build vector autoregressive (VAR) models and vector autoregressive-moving average (VARMA) models.
- **Modal parametric models**—For dynamic systems, especially those for which a bulk structural vibration model is useful, you can build models using the following modal parameters: natural frequencies, damping factors, resonance magnitudes, and resonance phases.
- **Stochastic state-space models**—For multivariate time series, you can build state-space models that characterize the dynamic behavior of a system.

Note The LabVIEW System Identification Toolkit also provides tools for building dynamic models. These tools focus on building dynamic models with both the stimulus and response signals of the system. The Time Series Analysis Tools focus on building models using only the response signals of the system.

Building Autoregressive Models

AR models of a time series enable you to predict the current value x_t of a time series, based on the past values x_{t-1}, x_{t-2},... x_{t-n}, plus a prediction error. The parameter n determines the number of past values you use to predict the current value. The following equation defines an AR model with an order of n:

$$x_t + a_1 x_{t-1} + a_2 x_{t-2} + \ldots + a_n x_{t-n} = e_t \quad (5\text{-}1)$$

where [1, a_1, a_2,... a_n] are the AR coefficients and e_t is the prediction error. Ideally, the residual prediction error is white noise with a mean value of zero.

You can rewrite Equation 5-1 more concisely as follows:

$$A(q)x_t = e_t$$

where $A(q)$ is the AR operator, which is defined as follows:

$$A(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + \ldots + a_n q^{-n}$$

The term q^{-k} is the backward shift operator, which is defined as follows:

$$q^{-k} x_t = x_{t-k} \quad (5\text{-}2)$$

From a dynamic-system point of view, a time series is the response of a linear system with a white noise as the stimulus. An AR model or other models of the response signal describe the linear system. The prediction error of the model is the white noise e_t. Figure 5-1 shows a diagram that uses the AR model to describe a linear system.

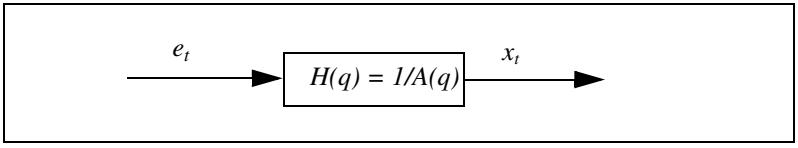

Figure 5-1. Signal Flow of an AR Model

H(q) represents the discrete-time transfer function of a physical system that generates the time series x_t. Because *H(q)* is an AR model, it has only poles and no zeroes. The roots of the polynomial *A(q)* are the poles of *H(q)*. Therefore, after you estimate the AR model of a time series, you can use the resulting AR coefficients to estimate the dynamic characteristics of the system that generates the time series.

Note If a time series is generated by a linear system with a stimulus other than white noise, the characteristics of the stimulus become part of the response time series. After you build an AR model for the response time series, the AR model reflects the characteristics of both the linear system and the non-white stimulus.

Because many real-world linear systems can be modeled accurately with AR models, AR models are a good first choice for parametric modeling. The computation of AR models also is particularly efficient because in contrast with MA and ARMA models, you only need to compute linear-regression equations. Furthermore, the resulting model is unique and stable. AR models are numerically preferable to ARMA models, especially when the model order is high. However, AR models may not accurately model linear systems that do not have an AR response, or in cases where the measured time series is contaminated with noise or distortion. If an AR model is not appropriate, a high model order may be required to whiten the residual prediction error e_t. But if you use a high model order to force an AR model to fit a particular time series for which an AR model is not appropriate, you may get spurious spectral components in the resulting response. Refer to Chapter 6, *Performing Correlation and Spectral Analysis*, for information about model-based spectral analysis methods.

For a multivariate time series with *m* variables, you can use an (m × 1)-length vector X_t to represent a multivariate time series, where $X_t^T = (x_{1t},... x_{mt})$. To describe the interrelationship between these variables, you can extend Equation 5-1 to be a VAR model as follows:

$$X_t + \boldsymbol{A_1} X_{t-1} + \boldsymbol{A_2} X_{t-2} + \ldots + \boldsymbol{A_n} X_{t-n} = \boldsymbol{E_t} \qquad (5\text{-}3)$$

where *n* is the model order, $\boldsymbol{I}$, $\boldsymbol{A_1}$, $\boldsymbol{A_2}$,... $\boldsymbol{A_n}$ are square matrices of the VAR coefficients, $\boldsymbol{I}$ is the identity matrix, the dimension of each matrix is $m \times m$, and $\boldsymbol{E_t}$ is the prediction-error vector, where $\boldsymbol{E_t}^T = (e_{1t}, e_{2t},... e_{mt})$. Each variable in $\boldsymbol{E_t}$ ideally is white noise with a mean value of zero. If the model fit is good, these variables are not correlated with each other.

You can rewrite Equation 5-3 concisely as follows:

$$\boldsymbol{A}(\boldsymbol{q})X_t = \boldsymbol{E_t}$$

where ***A(q)*** is the AR operator, which is defined as follows:

$$\boldsymbol{A}(\boldsymbol{q}) = 1 + \boldsymbol{A_1}q^{-1} + \boldsymbol{A_2}q^{-2} + \ldots + \boldsymbol{A_n}q^{-n}$$

The resulting VAR model of a multivariate time series is the coefficients matrix. However, you cannot obtain the dynamic characteristics of a multi-output system directly from the coefficients matrix of the VAR model. You must convert the model coefficients matrix into a state transition matrix in a stochastic state-space model. By computing the eigenvalues of a state transition matrix, you can obtain the poles of the system that generates the corresponding multivariate time series. You then can obtain the dynamic characteristics of the system from the poles.

Use the TSA AR Modeling VI to build AR and VAR models for univariate or multivariate time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Selecting an Appropriate AR Order

The first step of estimating a model is to select an appropriate model. For a given model, selecting the model order is typically a trial-and-error process. Besides using background knowledge about the physical system that generates the time series, you also need to use other information, such as the information acquired from various statistical analysis methods, to justify the selected model order.

One tool for determining the model order is the partial auto-correlation function of the time series. The partial auto-correlation function is a function of lag. The partial auto-correlation value becomes very small when the lag equals a suitable AR order. Figure 5-2 shows an example of estimating the AR order with the partial auto-correlation function.

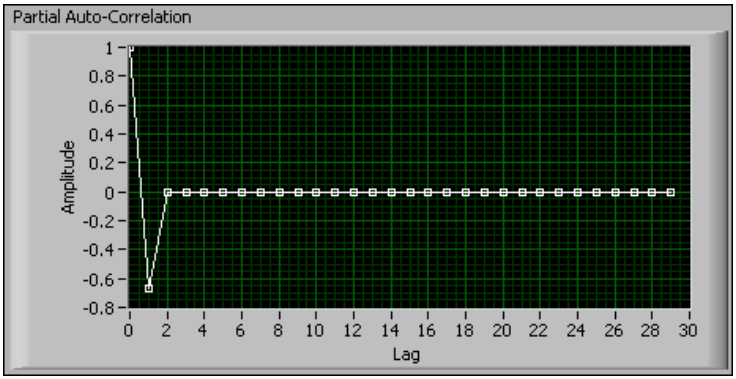

Figure 5-2. AR Order Estimation with the Partial Auto-Correlation Function

The value of the **Partial Auto-Correlation** plot in Figure 5-2 becomes zero when lag equals two or greater. Therefore, a suitable AR order for this model is two. Refer to the *Correlation Methods* section of Chapter 6, *Performing Correlation and Spectral Analysis*, for information about the correlation methods.

Instead of computing the partial auto-correlation function for a time series, you can use a set of model-selection criteria to estimate the model order. From a least-square fitting standpoint, the higher the model order, the better the model fits the time series, because a high-order model has more degrees of freedom. However, an unnecessarily high-order may introduce spurious spectral artifacts in the resulting response. As a result, the criteria you use to assess the model order therefore must not only rely on the model-fitting error but also incorporate a penalty when the order increases. Different selections for the penalty determine different criteria.

Akaike's Information Criterion

The Akaike's Information Criterion (AIC) is a weighted estimation error based on the unexplained variation of a given time series with a penalty term when exceeding the optimal number of parameters to represent the system. For the AIC, an optimal model is the one that minimizes the following equation:

$$AIC = V_n\left(1 + \frac{2n}{L}\right)$$

where L is the number of data points in a time series, n is the model order, and V_n is the prediction error.

Bayesian Information Criterion

The Bayesian Information Criterion (BIC) replaces the term $2n$ in the AIC with the expression $(n + nln(L))$. The BIC penalizes excess model order more severely than the AIC does. For the BIC, an optimal model is the one that minimizes the following equation:

$$BIC = V_n\left(1 + \frac{n + n\ln L}{L}\right)$$

Final Prediction Error Criterion

The Final Prediction Error Criterion (FPE) estimates the model-fitting error when you use the model to predict new outputs. For the FPE, an optimal model is the one that minimizes the following equation:

$$FPE = V_n\left(1 + \frac{2n}{L - n}\right)$$

Minimal Description Length Criterion

The Minimal Description Length Criterion (MDL) is based on V_n plus a penalty for the number of terms used. For the MDL, an optimal model is the one that minimizes the following equation:

$$MDL = V_n\left(1 + \frac{n\ln L}{L}\right)$$

Phi Criterion

The Phi Criterion (PIC) generates an optimal model that minimizes the following equation:

$$PHI = V_n\left(1 + \frac{2n\ln(\ln L)}{L}\right)$$

Use the TSA AR Modeling Order VI to estimate a suitable AR model order for a time series. This VI implements the partial auto-correlation function and uses the AIC, BIC, FPE, MDL, and PIC methods to search for the optimal model order in the range of interest. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

In the **Browse** tab of the NI Example Finder, you can find an AR order-estimation example by selecting **Toolkits and Modules»Time Series Analysis»Getting Started»AR Order Estimation VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Time Series Analysis*, for information about launching the NI Example Finder.

Building Autoregressive-Moving Average Models

ARMA models are more accurate than AR models in approximating the response of linear systems with zeroes in the response transfer function. The following equation defines an ARMA model with an AR order of n and the MA order of m:

$$x_t + a_1 x_{t-1} + a_2 x_{t-2} + \ldots + a_n x_{t-n} = e_t + b_1 e_{t-1} + \ldots + b_m e_{t-m} \quad (5\text{-}4)$$

where $[1, a_1, a_2, \ldots a_n]$ are the AR coefficients, and $[1, b_1, b_2, \ldots b_m]$ are the MA coefficients.

You can rewrite Equation 5-4 more concisely as follows:

$$A(q)x_t = B(q)e_t \quad (5\text{-}5)$$

where $A(q)$ is the AR operator and $B(q)$ is the MA operator defined as follows:

$$B(q) = 1 + b_1 q^{-1} + b_2 q^{-2} + \ldots + b_m q^{-m}$$

Figure 5-3 shows a diagram that uses an ARMA model to describe a linear system. You can consider a time series x_t as being the response of a system whose transfer function $H(q)$ is $B(q)/A(q)$. The stimulus of the system is e_t.

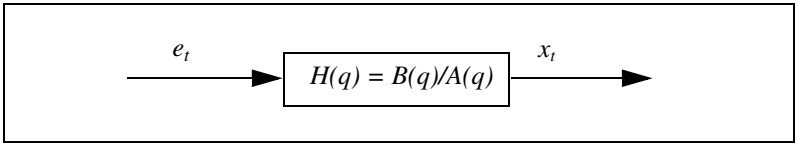

Figure 5-3. Signal Flow of an ARMA Model

$H(q)$ contains both zeroes and poles. The roots of the polynomial $A(q)$ are the poles of $H(q)$. The roots of the polynomial $B(q)$ are the zeroes of $H(q)$. The ARMA model uses poles and zeroes to describe a system. Therefore, comparing with the AR model, the ARMA model can give a more accurate description of the dynamic characteristics of a physical system that generates time series, if the system has an ARMA response. However,

because the MA term introduces nonlinearities in the model estimation, you cannot use an analytic form to solve the model coefficients. An iterative nonlinear optimization procedure is required, which could find a sub-optimal solution erroneously instead of finding the globally optimal solution.

You also can estimate a VARMA model for a multivariate time series. The resulting VARMA model of a multivariate time series is the coefficients matrix. However, you cannot obtain the dynamic characteristics of a multi-output system directly from the coefficients matrix of the VARMA model. You must convert the model coefficients matrix into a state transition matrix in a stochastic state-space model. By computing the eigenvalues of a state transition matrix, you can obtain the poles of the system that generates the corresponding multivariate time series. You then can obtain the dynamic characteristics of the system from the poles.

An MA model is a special case of the ARMA model that does not contain poles. The MA model describes a time series according to the following equation:

$$x_t = B(q)e_t$$

Use the TSA MA Modeling VI to build MA and VMA models for univariate or multivariate time series, respectively. Use the TSA ARMA Modeling VI to build ARMA and VARMA models. You also can use the TSA ARMA Modeling VI to build MA models by setting the AR order to zero. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about these VIs.

Note An AR model also is a special case of the ARMA model with the MA order equals zero. However, the TSA ARMA Modeling VI is not suitable for estimating AR models. Use the TSA AR Modeling VI to estimate an AR model, because the TSA AR Modeling VI is more efficient than the TSA ARMA Modeling VI.

Selecting an Appropriate ARMA Order

Before estimating the ARMA model coefficients of a time series, you need to specify a suitable order for the model. Generally, you specify an AR order of n and an MA order of $n - 1$, where n is an integer. If you know the order of the response of the system that generates the time series, you can use that order directly. If you do not know the order, you can try orders from low to high to identify the lowest order in which the prediction error is sufficiently white. You can use the TSA Whiteness Test VI to perform whiteness tests on the residual prediction error, by computing

auto-correlation and the corresponding confidence limit for the auto-correlation. Refer to the *Checking the Stationarity and Whiteness* section of Chapter 4, *Performing Statistical Analysis*, for information about performing whiteness tests.

The variance of the prediction error also indicates if the ARMA model at a specified order fits the time series well. The smaller the variance of the prediction error is, the better the estimated model fits the time series. However, the variance of the prediction error for ARMA models decreases monotonically with order. Unnecessarily high orders introduce spurious artifacts in the resulting response if the measured time series is contaminated with noise or distortion.

The changes in the variance of the prediction error that result from changes in the model order can indicate dynamic characteristics of a physical system that generates time series. This property is helpful in vibration monitoring applications.

In the **Browse** tab of the NI Example Finder, you can find a machinery monitoring example by selecting **Toolkits and Modules»Time Series Analysis»Applications»Engine Knocking Monitoring VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Time Series Analysis*, for information about launching the NI Example Finder.

Building Modal Parametric Models

A modal parametric model describes the bulk dynamic characteristics of resonant systems such as buildings and bridges. The dynamic characteristics include natural frequencies, damping factors, resonance magnitudes, and resonance phases.

You can use the impulse response function to determine the response of a linear system to stimulus. The impulse response time series for a linear system can be represented parametrically with a linear combination of damped complex sinusoids as follows:

$$h_t = \sum_{i=1}^{n} g_i e^{s_i t}$$

where t is the time index, n is the model order, g_i is one of the complex magnitudes, and s_i is one of the modal poles. s_i contains a unified frequency f_i and a damping factor α_i as follows:

$$s_i = \alpha_i + j2\pi f_i$$

The parameter g_i contains a magnitude r_i and a phase θ_i as follows:

$$g_i = r_i e^{j\theta_i}$$

So a discrete-time system can be defined by a series of resonance components, or modes: natural frequencies f_i, damping factors α_i, resonance magnitudes r_i, and resonance phases θ_i.

For a multivariate impulse response time series, you can build a modal parametric model as follows:

$$H_t = \sum_{i=1}^{n} G_i e^{s_i t}$$

where $\boldsymbol{H_t}$ is a $(k \times 1)$ vector with k variables that come from k sources. $\boldsymbol{H_t^T} = [h_{1t}, h_{2t}, \ldots h_{kt}]$. $\boldsymbol{G_i}$ is a $(k \times 1)$ complex magnitude vector with k variables. $\boldsymbol{G_i^T} = [g_{1i}, g_{2i}, \ldots g_{ki}]$. Each variable in the vector $\boldsymbol{G_i}$ is one of the complex magnitudes. s_i is one of the modal poles.

Specifying an Appropriate Model Order

The model order determines the number of modes that a modal parametric model contains. A model can contain real modes or pairs of complex conjugate modes. A real mode generates a resonance component at 0 Hz or at the Nyquist frequency. A pair of conjugate complex modes generates a resonance component with a positive frequency and discards the conjugate resonance component with the corresponding negative frequency. To search for m positive resonance components, you must specify the model order to be at least $2 \times m$. If a time series contains a large offset, you need to set the model order to at least $2 \times m + 1$ to allow for a real mode. Structural vibration time series typically do not have offsets, so you can use a model order of $2 \times m$.

Note Noise and distortion in the measured time series, as well as an insufficient model order, can result in the estimated model containing spurious components. To discern which components in the estimated model are valid, you can try specifying different model orders and form a histogram for the estimated resonance components. In the histogram, the true resonance components of the dynamic system typically do not change with model orders. However, spurious resonance components may change with orders, and weak resonance components also may change with orders, especially if the measured time series is noisy

or distorted. In the **Browse** tab of the NI Example Finder, you can find a modal analysis example by selecting **Toolkits and Modules»Time Series Analysis»Applications» Modal Analysis of a Beam VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Time Series Analysis*, for information about launching the NI Example Finder.

Use the TSA Modal Parametric Modeling VI to build modal parametric models for univariate or multivariate time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Figure 5-4 shows an example of computing the modes of a synthesized univariate time series consisting of two sinusoids at the frequencies of 120 Hz and 130 Hz.

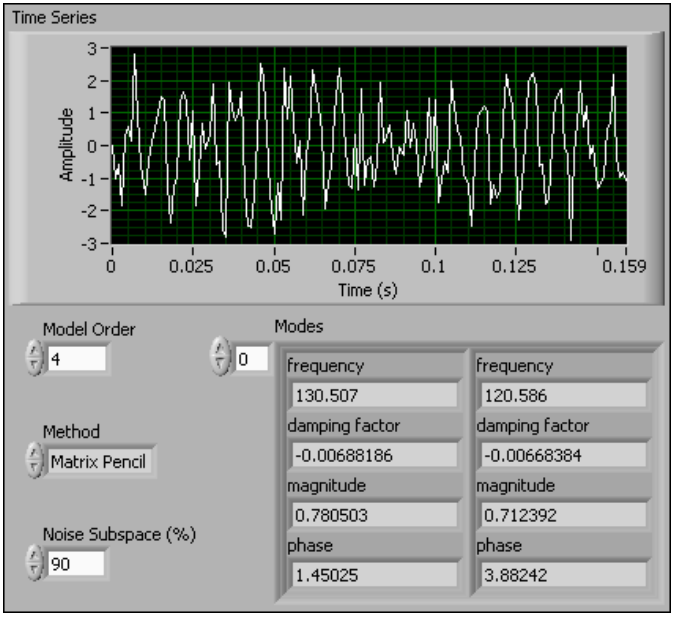

Figure 5-4. Estimating Modes of a Time Series

This example uses the **Matrix Pencil** method because this method is less sensitive to noise than the other available methods. The value of **Model Order** is 4, which equals twice the number of the sinusoids in the univariate time series.

The **Noise Subspace** value is set to 90% to compensate for additive noise in the measurements. For a fixed number of resonance components, a large dimension for the **Noise Subspace** parameter results in a large dimension for the signal subspace you use to describe a time series. A large dimension for the signal subspace helps you reduce the modeling error for the time series. However, an excessively large dimension may introduce spurious resonance components if the time series does not contain much noise. In Figure 5-4, the **Modes** array accurately indicates the attribute of the resonance components that the synthesized time series contains.

Building Stochastic State-Space Models

A stochastic state-space model describes an output-only dynamic system according to the following equations:

$$x_{k+1} = \boldsymbol{A}x_k + w_k$$
$$y_k = \boldsymbol{C}x_k + v_k$$

where y_k is the $(m \times 1)$ multivariate time series with m variables, x_k is the state vector with n state variables, n is the model order, $\boldsymbol{A}$ is the state transition matrix with a dimension of $n \times n$, $\boldsymbol{C}$ is the measurement matrix, or state observation matrix, with a dimension of $m \times n$, and w_k and v_k are the $(n \times 1)$ and $(m \times 1)$ noise vectors with a mean value of zero, respectively.

The matrix $\boldsymbol{A}$ is defined as follows:

$$\boldsymbol{A} = \Psi\Lambda\Psi^{-1}$$

where Λ is a diagonal matrix containing the complex eigenvalues λ_i $(i = 1, 2,... n)$ of matrix $\boldsymbol{A}$. Ψ contains the eigenvectors as columns.

You can obtain the modal pole s_i from the complex eigenvalues λ_i as follows:

$$s_i = \ln(\lambda_i)$$

The eigenvalues of the state transition matrix $\boldsymbol{A}$ characterize the dynamic behavior of a physical system. By computing the state transition matrix $\boldsymbol{A}$ and measurement matrix $\boldsymbol{C}$, you also can obtain the following modal

parameters defined in the modal parametric model: natural frequencies, damping factors, resonance magnitudes, and resonance phases. Refer to the *Building Modal Parametric Models* section of this chapter for more information about the modal parameters defined in the modal parametric model.

You can obtain the complex magnitude $\boldsymbol{G_i}$ vector for each modal pole s_i from the measurement matrix $\boldsymbol{C}$, using the state observation equation as follows:

$$\Phi = \boldsymbol{C}\Psi$$

Each column of the matrix Φ is associated with one complex magnitude $\boldsymbol{G_i}$.

Use the TSA Stochastic State-Space Modeling VI to build stochastic state-space models for multivariate time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Using the Time Series Modeling Express VI

You can use the Time Series Modeling Express VI to build the following models for a univariate or multivariate time series interactively:

- AR and VAR models
- MA and VMA models
- ARMA and VARMA models

Figure 5-5 shows the configuration dialog box of the Time Series Modeling Express VI.

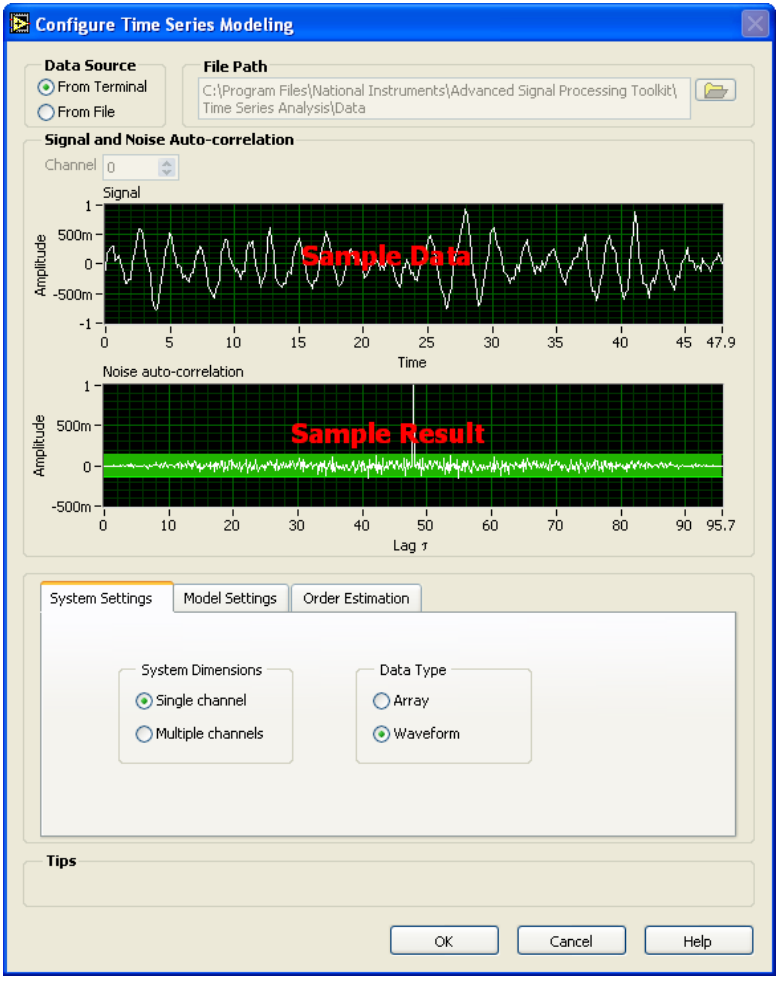

Figure 5-5. Configure Time Series Modeling Dialog Box

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this Express VI.

To build an AR model for a univariate time series, complete the following steps:

1. Place the Time Series Modeling Express VI on the block diagram. The **Configure Time Series Modeling** dialog box, as shown in Figure 5-5, automatically launches.

2. In the **Data Source** section of the configuration dialog box, specify if the time series is from a terminal of another VI or from a pre-stored file.
3. If you specify **From Terminal** as the data source, complete the following steps:
   a. On the **System Settings** page, select **Single channel** for **System Dimensions** and specify an appropriate data type for the univariate time series in **Data Type**.
   b. Click the **OK** button to close the configuration dialog box.
   c. Connect the univariate time series to the express VI. Click the **Run** button on the block diagram.
   d. Double-click the Time Series Modeling Express VI to reopen the **Configure Time Series Modeling** dialog box. The **Signal** graph displays the loaded time series.
4. Click the **Model Settings** tab and select **AR** in **Type**.
5. Click the **Order Estimation** tab. On the **Order Estimation** page, complete the following steps to estimate a suitable AR order for the time series:
   a. In the **Method** list, select an appropriate criterion function.

      Refer to the *Selecting an Appropriate AR Order* section of this chapter for information about the estimation criteria.
   b. Specify an appropriate AR order range in the **Maximum AR order** box and the **Minimum AR order** box.
   c. Click the **Estimate** button.

      This Express VI computes the value of the specified criterion function within the order range and highlights the optimal order in the **Criterion Function** graph, as shown in Figure 5-6. You also can select a different order manually on the **Criterion Function** graph.

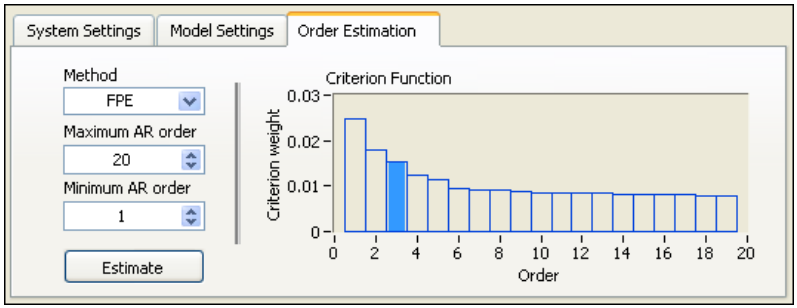

Figure 5-6. Order Estimation Page

6. Click the **Model Settings** tab. In the **AR order** box, specify the value estimated in Step 5. In the **Method of AR** list, specify an appropriate estimation method.
7. Verify the estimated model by watching the **Noise auto-correlation** plot of model error. If the estimated model is suitable for the univariate time series, the values of the auto-correlation without lag *t* being zero fall into the confidence interval.
8. Click **OK** to close the configuration dialog box.

   You now can use this configured Express VI to estimate the AR model for a univariate time series.

Predicting Time Series Values

You can predict future values of a time series using the Time Series Analysis Tools, which provide the following two types of prediction methods:

- **ARMA Model-based prediction**—You can make predictions based on the estimated ARMA model for both univariate and multivariate time series.
- **Exponential smoothing prediction**—You can make predictions based on exponential smoothing for univariate time series.

ARMA Model-Based Prediction

The ARMA model is a general model that includes both the AR and the MA models. This section describes predictions based only on ARMA models. You can apply ARMA model-based prediction to AR or MA model-based prediction by converting an AR or MA model to an ARMA model.

Use the TSA ARMA Prediction VI to predict the future values of univariate or multivariate time series based on the estimated ARMA or VARMA models. Refer to the *LabVIEW Help*, available by selecting **Help» VI, Function, & How-To Help**, for information about this VI.

Figure 5-7 shows an example of predicting the monthly temperatures for the following year based on the ARMA model of the monthly temperatures during the previous eleven years.

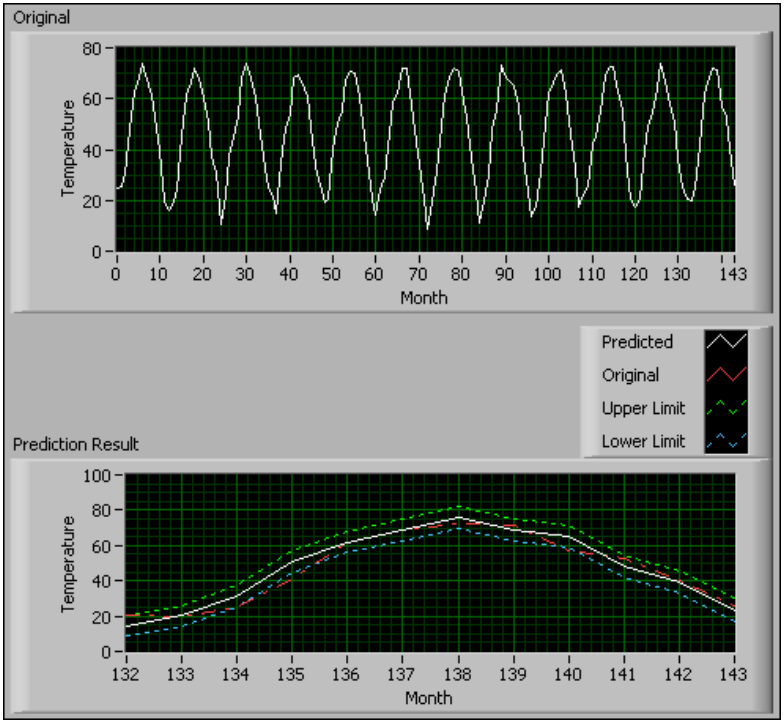

Figure 5-7. Model-Based Prediction on Monthly Temperatures[1]

In Figure 5-7, the **Original** graph contains the monthly average temperatures for twelve consecutive years. Before predicting future values, this example estimates the AR model for the temperature time series of the first eleven years. With the estimated AR model coefficients and by specifying the MA model coefficients to one, this example predicts the monthly temperatures for the twelfth year using the TSA ARMA Prediction VI.

[1] Data source: Hyndman, R.J. (n.d.) *Time Series Data Library*, `http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/`

The **Prediction Result** graph in Figure 5-7 compares the predicted temperatures with the original temperatures of the twelfth year by plotting them together. The prediction may bias from the true values but the true values fall into the estimated confidence range. The **Upper Limit** and **Lower Limit** plots in the **Prediction Result** graph indicate the confidence range of the prediction. You can see in the **Confidence Level** indicator that the prediction result falls into the estimated confidence range with the confidence level of 99.73% (3σ).

In the **Browse** tab of the NI Example Finder, you can view this example by selecting **Toolkits and Modules»Time Series Analysis» Getting Started»ARMA Prediction VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Time Series Analysis*, for information about launching the NI Example Finder.

Exponential Smoothing Prediction

Exponential smoothing prediction builds an exponential smoothing model for a time series and then predicts the future values of the time series based on the model. Exponential smoothing prediction is suitable for a time series that contains trends and seasonal variations.

Use the TSA Exponential Prediction VI to predict the future values of univariate or multivariate time series based on exponential smoothing. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Figure 5-8 shows an example of performing exponential smoothing prediction on a simulated sales record.

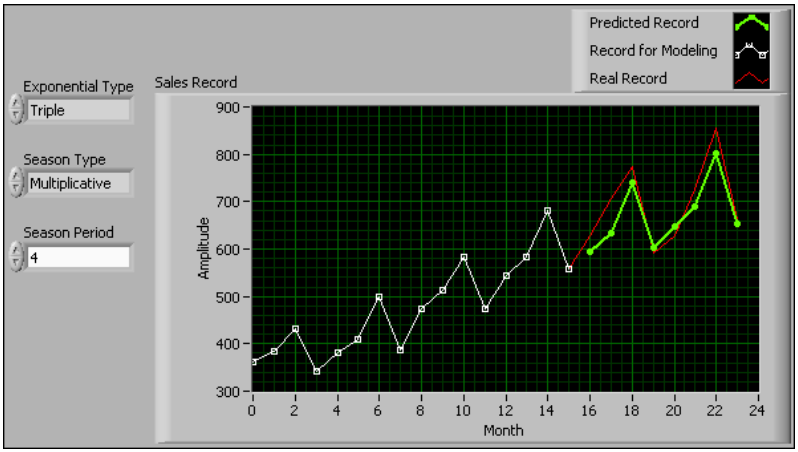

Figure 5-8. Exponential Prediction on a Simulated Sales Record

The **Sales Record** graph contains a univariate time series of a sales record for a certain product. This example splits the time series into two parts—one part for modeling and the other part for comparison—and specifies the following settings for prediction:

- Sets **Exponential Type** to **Triple** because the sales record contains a systematic trend and seasonality.
- Sets **Season Type** to **Multiplicative** because the amplitude of the sales record increases over time.
- Sets **Season Period** to 4 because the sales record contains four points in each oscillation period.

The **Prediction Result** graph in Figure 5-8 shows the **Predicted Record** plot matches the **Real Record** plot.

In the **Browse** tab of the NI Example Finder, you can view this example by selecting **Toolkits and Modules»Time Series Analysis» Getting Started»Exponential Prediction VI**. Refer to the *Finding Example VIs* section of Chapter 1, *Introduction to Time Series Analysis*, for information about launching the NI Example Finder.

6

Performing Correlation and Spectral Analysis

One application of time series analysis is detecting the hidden periodicities or frequency characteristics in a time series at specific frequencies. Using the LabVIEW Time Series Analysis Tools, you can detect a periodicity of a time series in either the time domain or the frequency domain.

In the time domain, correlation analysis is a classical method that you can use to find periodic patterns at a specific frequency in one or more time series. You also can use the correlation method to identify or extract other useful features of a time series, such as phase.

In the frequency domain, you can use spectral analysis methods to estimate the power spectrum of a time series. You also can estimate the bispectrum, which is useful for detecting nonlinearities in a time series, and the cepstrum, which is useful for deconvolving a time series.

This chapter provides information about the correlation analysis methods and various power spectrum estimation methods that you can perform with the Time Series Analysis Tools.

Correlation Methods

Correlation measures the similarity between two signals. The correlation methods include auto-correlation and cross-correlation. Auto-correlation is the correlation of a time series with itself. You can use the auto-correlation method to capture periodic components in a univariate time series without other reference time series. The auto-correlation operation preserves the frequency and amplitude information of periodic components but does not preserve the phase information.

Figure 6-1 shows two noise-contaminated, phase-jittered time series and the resulting auto-correlation plots. In the **Auto-Correlation 1** and **Auto-Correlation 2** graphs, you can see that auto-correlation greatly attenuates the noise in the original time series, thereby potentially

improving the accuracy and reliability of subsequent frequency or periodicity estimates calculated for the time series.

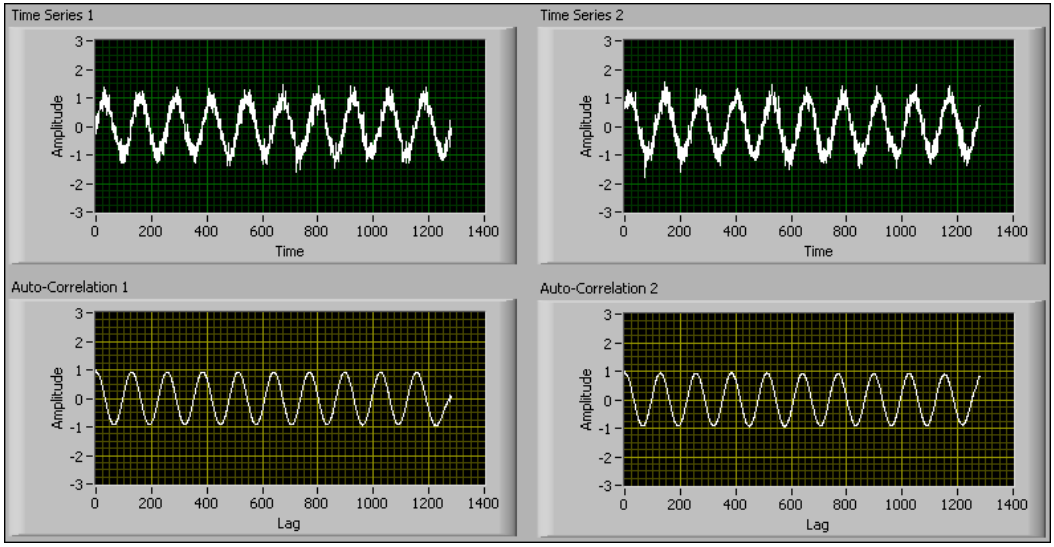

Figure 6-1. Noise-Contaminated Time Series and the Auto-Correlations

As Figure 6-1 shows, though the time series in the **Time Series 1** and **Time Series 2** have a difference in phase, the auto-correlations are close to each other. Therefore, auto-correlation does not preserve the phase information, which can be a beneficial property in applications such as feature extraction.

When performing auto-correlation of a time series, you can choose either biased or unbiased weighting. Biased weighting is helpful when you perform spectral analysis with the Fourier transform, because biased weighting can reduce spectral leakage, bias, and variance in the resulting power spectrum. Unbiased weighting is suitable for periodicity detection in the time domain. If you use unbiased weighting, the amplitudes of the auto-correlation values do not decrease as the lag increases.

Figure 6-2 shows a time series of a sinusoidal pattern.

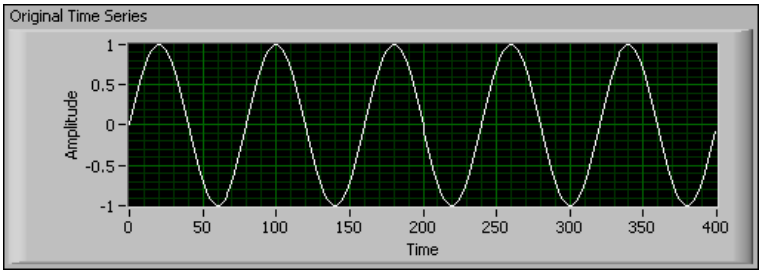

Figure 6-2. Time Series with a Sinusoidal Pattern

Figure 6-3 shows the estimated biased and unbiased auto-correlation of the sinusoidal pattern, respectively.

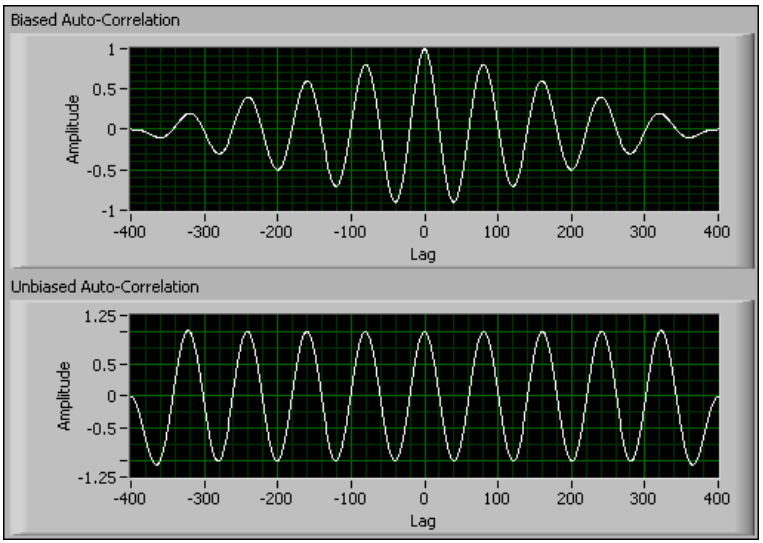

Figure 6-3. Biased and Unbiased Auto-Correlations of the Sinusoidal Pattern

Use the TSA Auto-Correlation Function VI to perform auto-correlation operations on a univariate time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

You also can perform partial auto-correlation on a time series using the TSA Auto-Correlation Function VI. Partial auto-correlation is useful for estimating the autoregressive (AR) model order of a time series. Refer to the *Selecting an Appropriate AR Order* section of Chapter 5, *Building*

Models and Predicting Time Series Values, for information about estimating an appropriate AR order using the partial auto-correlation function.

Cross-correlation measures the similarity between two time series. You can find the features of an unknown time series by computing the cross-correlation between the unknown time series and a known time series. For example, if you know the frequency of a sinusoid time series but do not know the amplitude and phase, you can derive the amplitude and phase of the time series with the cross-correlations. Figure 6-4 shows the flowchart of obtaining the magnitude response and phase response of a physical system with the cross-correlation method.

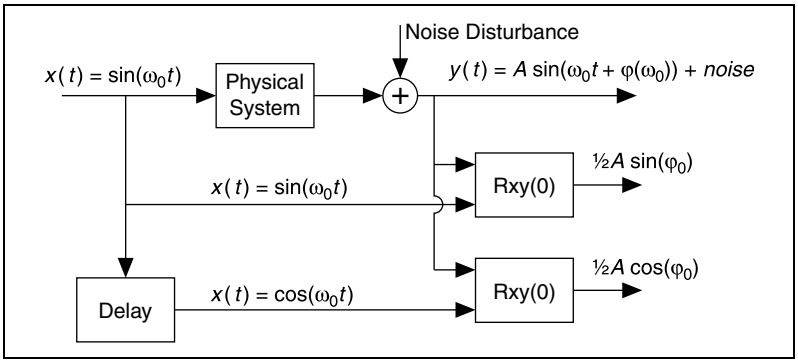

Figure 6-4. Cross-Correlation Flowchart

In this flowchart, a sinusoid stimulus signal $x(t) = \sin(\omega_0 t)$ at a specified frequency ω_0 is applied to the physical system. The system then generates the response signal $y(t) = A(\omega_0)\sin(\omega_0 t + \varphi(\omega_0))$. $A(\omega_0)$ and $\varphi(\omega_0)$ define the magnitude response and phase response of the physical system at ω_0, respectively. The term $1/2A\sin(\varphi_0)$ is obtained by performing cross-correlation between the stimulus signal $x(t)$ and the response signal $y(t)$ at $\tau = 0$, where τ denotes the lag. The term $1/2A\cos(\varphi_0)$ is obtained by performing cross-correlation between the phase-delayed stimulus signal $x(t) = \cos(\omega_0 t)$ and the response signal $y(t)$ at $\tau = 0$. You can obtain the magnitude response $A(\omega_0)$ from the square root of $(1/2A\sin(\varphi_0))^2 + (1/2A\cos(\varphi_0))^2$. You can obtain the phase response $\varphi(\omega_0)$ from the inverse tangent of the quotient of $1/2A\sin(\varphi_0)$ divided by $1/2A\cos(\varphi_0)$. If you change the frequency of the stimulus signal, you can obtain the response of the system at other frequencies.

Use the TSA Cross-Correlation Function VI to perform cross-correlation on two time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Power Spectrum Estimation Methods

A power spectrum describes the energy distribution of a time series in the frequency domain. Energy is a real-valued quantity, so the power spectrum does not contain phase information. Because a time series may contain non-periodic or asynchronously-sampled periodic signal components, the power spectrum of a time series typically is considered to be a continuous function of frequency. When you use a series of discrete frequency bins to represent the continuous frequency, the value at a specific frequency bin is proportional to the frequency interval. To remove the dependence on the size of the frequency interval, you can normalize the power spectrum to produce the power spectral density (PSD), which is the power spectrum divided by the size of the frequency interval.

The PSD measures the signal power per unit bandwidth for a time series in V^2/Hz, which implicitly assumes that the PSD represents a signal in volts driving a 1 ohm load. If the PSD is represented in a decibel (dB), the corresponding unit for the PSD is dB ref V/sqrt(Hz). If you want to use other units for the estimated PSD of a time series, you need to scale the unit of the time series into appropriate engineering units (EU). After scaling the unit of the time series, you can obtain the corresponding unit for the linear PSD value and the dB PSD value as EU^2/Hz and dB ref EU/sqrt(Hz), respectively. Use the TSA Scale to EU VI to scale the unit for a time series to appropriate EU. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

PSD estimation methods are classified as follows:

- **Parametric methods**—These methods are based on parametric models of a time series, such as AR models, moving average (MA) models, and autoregressive-moving average (ARMA) models. Therefore, parametric methods also are known as model-based methods. To estimate the PSD of a time series with parametric methods, you need to obtain the model parameters of the time series first.

  You must build an appropriate model that correctly reflects the behavior of the system that generates the time series; otherwise, the estimated PSD might not be reliable. Refer to Chapter 5, *Building Models and Predicting Time Series Values*, for information about building models for a time series.

  The multiple signal classification (MUSIC) method also is a model-based spectral estimation method. Refer to the *Using the MUSIC Method* section of this chapter for information about this method.

- **Nonparametric methods**—These methods, which include the periodogram and the Welch methods, are based on the discrete Fourier transform. You do not need to obtain the parameters of the time series first.

The primary limitation of nonparametric methods is that the computation uses data windowing, resulting in distortion of the resulting PSDs due to window effects. The key benefit of nonparametric methods is the robustness—the estimated PSDs do not contain spurious frequency peaks. In contrast, parametric methods do not use data windowing. Parametric methods assume a signal fits a particular model. The estimated PSDs may contain spurious frequency peaks if the assumed model is wrong. PSDs estimated with parametric methods are less biased and possess a lower variance than PSDs estimated with nonparametric methods if the assumed model is correct. However, the magnitudes of PSDs estimated with parametric methods usually are incorrect.

Note During spectral analysis, you can average successive spectrum measurements to reduce estimation variance and improve measurement accuracy. Use the TSA Average PSD VI to average the estimated spectrum continuously. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Using the Model-Based Methods

Model-based methods assume that a time series is the response from a linear system stimulated by white noise, as shown in Figure 6-5, where white noise $\boldsymbol{e_t}$ is the stimulus, $H(z)$ is the linear system, and $\boldsymbol{x_t}$ is the response time series. Correspondingly, the PSD of the response time series is the frequency response function (FRF) of the linear system.

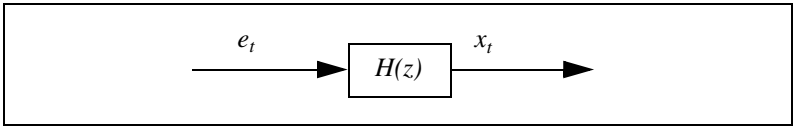

Figure 6-5. Model-Based Spectrum Estimation

In general, you can describe a linear system $H(z)$ with an AR, MA or ARMA model. To create such a model for a time series, you can estimate the model coefficients first and then use the estimated model coefficients to compute the PSD of the time series. For example, the PSD based on an ARMA model is computed by the following equation:

$$PSD(f) = \frac{\sigma^2 \left| FFT(b_k) \right|}{\left| FFT(a_k) \right|} \times \frac{1}{\Delta f} \qquad (6\text{-}1)$$

where σ^2 is the noise variance of the estimated ARMA model of a time series, a_k is the AR coefficients of the ARMA model that define the poles of the model, b_k is the MA coefficients of the ARMA model that define the zeroes of the model, and Δf is the frequency interval of the PSD.

The AR and MA models are the subsets of the ARMA model, so you also can compute the PSD of a time series based on the AR or MA model from Equation 6-1 by setting the corresponding model coefficients a_k or b_k to zero.

Use the TSA AR Spectrum VI and the TSA ARMA Spectrum VI to compute the PSD of a time series with model-based methods. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about these VIs.

The PSD based on each model has specific characteristics. Figure 6-6 shows an AR model-based PSD for a signal from a system with two pairs of poles, which cause the peaks in the PSD at about 0.17 and 0.21 Hz, and one pair of zeroes, which cause the valley in the PSD at about 0.25 Hz. In this figure, the AR model gives a good estimation for peaks and a poor estimation for valleys in the PSD.

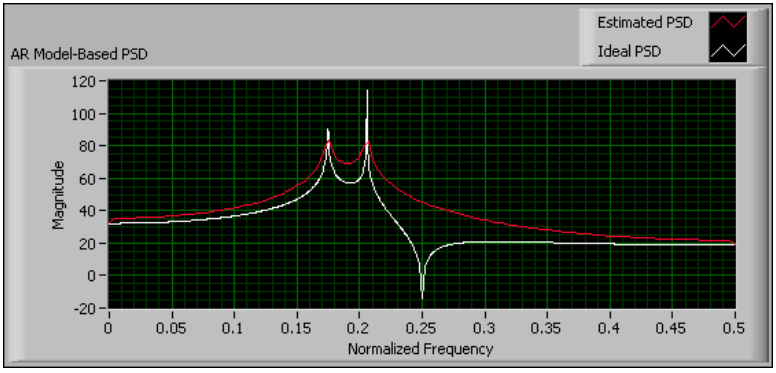

Figure 6-6. AR Model-Based PSD

Figure 6-7 shows an MA model-based PSD, which gives a good estimation for valleys and a poor estimation for peaks.

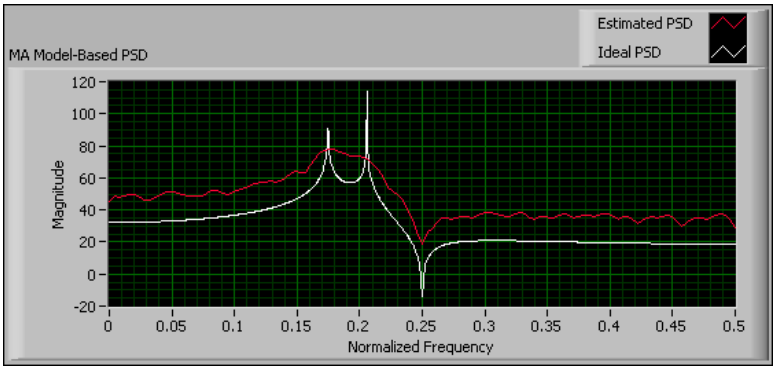

Figure 6-7. MA Model-Based PSD

Figure 6-8 shows an ARMA model-based PSD, which gives the best overall estimation.

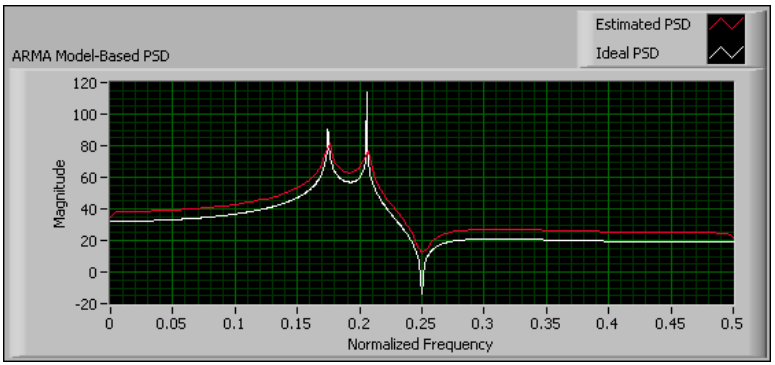

Figure 6-8. ARMA Model-Based PSD

In parametric modeling applications, you usually use the AR model-based method instead of the MA or ARMA methods to estimate the PSD of a time series, because this method computes the model coefficients through a linear equation. For real-valued time series, the AR model order is set to at least twice the number of frequency peaks that you want to analyze. The estimated PSD shows more details about the frequency peaks with a large AR model order; however, a much higher AR model order can lead to spurious peaks in the resulting PSD.

Using the MUSIC Method

The multiple signal classification (MUSIC) method is a model-based spectral estimation method. Assuming that a time series can be represented by a series of complex sinusoids with additive white noise, this method first computes the correlation matrix R_p of the time series and obtains the eigenvectors V_1, V_2,... V_p and eigenvalues λ_1, λ_2,... λ_p. The eigenvectors V_1, V_2,... V_M with large eigenvalues form the signal subspace. The remaining eigenvectors V_{M+1}, V_{M+2},... V_p form the noise subspace. The signal subspace also can be represented by group vectors of complex sinusoids $e(f_1)$, $e(f_2)$,... $e(f_M)$. The vector of a complex sinusoid is defined as follows:

$$e(f_i) = [1, e^{j2\pi f_i}, \ldots e^{j2\pi f_i M}]$$

If a time series contains a frequency component at f_i, the vector $e(f_i)$ is uncorrelated with V_{M+1}, V_{M+2},... V_p. Accordingly, the following equation generates a peak value at f_i:

$$PSD(f_i) = \frac{1}{\sum_{k=M+1}^{p} \left| e^T(f_i) V_k \right|^2} \times \frac{1}{\Delta f}$$

Note A PSD normally gives a good indication of the attributes of spectral components contained in a time series. However, the peaks in the PSD computed with the MUSIC method just indicate the frequency locations of components in a time series. The magnitude of each peak does not indicate the spectral power at the corresponding frequency.

A modified MUSIC method uses the eigenvalues λ_1, λ_2,... λ_p as a weighting vector to compute the PSD as follows:

$$PSD(f_i) = \frac{1}{\sum_{k=M+1}^{p} \left| \frac{1}{\lambda_k} e^T(f_i) V_k \right|^2} \times \frac{1}{\Delta f}$$

The modified MUSIC method, which also is called the Eigenvector method, can reduce the variance in the estimated PSD.

When using the MUSIC method, you must specify the size of the noise subspace, which is defined as $p - M$. In general, you can specify a rough percentage of the whole space for the size of the noise subspace. With a large size of the noise subspace, you can obtain a smooth PSD but the

resulting PSD may miss weak spectral peaks in the signal. With a small size of the noise subspace, you can obtain a detailed PSD that reveals weak spectral peaks. However, a too small size for the noise subspace leads to spurious peaks in the resulting PSD.

Use the TSA MUSIC VI to compute the PSD of a time series with the MUSIC method. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Using the Periodogram Method

The periodogram method is the most common nonparametric method for computing the PSD of a time series. This method computes the PSD with the fast Fourier transform (FFT) according to the following equation:

$$PSD(f) = \frac{\left|FFT(W_t \times X_t, f)\right|^2}{L} \times \frac{1}{\Delta f}$$

where X_t a time series, L is the number of samples in the time series, W_t is the applied window function, and Δf is the frequency interval.

The applied window function reduces spectral leakage in the estimation, but the window function also decreases the frequency resolution at the same time. Refer to the *LabVIEW Analysis Concepts* for information about balancing spectral leakage and frequency resolution with an appropriate window function.

Use the TSA Periodogram VI to compute the PSD of a time series. If a time series contains non-periodic signal components or periodic components that are not sampled synchronously, the value of the resulting PSD computed by this VI may have a large variance at each frequency. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Using the Welch Method

The Welch method reduces the variance of the periodogram method by averaging. This method first divides a time series into overlapping subsequences by applying a window to each subsequence and then averaging the periodogram of each subsequence.

The length of the applied window controls the trade-off between bias and variance of the resulting PSD. Figure 6-9 shows the resulting PSD for a signal from a system with two pairs of poles at about 0.17 and 0.21 Hz,

when the window length is 2048 and 128 for the top and bottom plots, respectively.

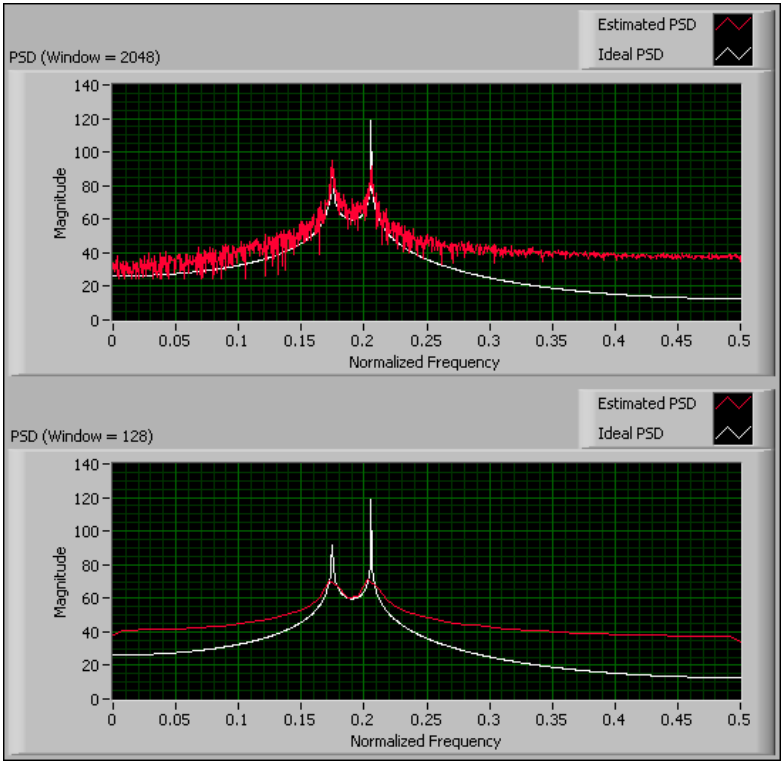

Figure 6-9. Estimated PSD with Different Window Lengths

In the **PSD (Window = 2048)** graph, a large window generates a PSD with small bias in the locations of the two peaks, as you can see that the magnitudes of the peaks in the **Estimated PSD** plot almost equal the magnitudes of the peaks in the **Ideal PSD** plot. However, a large window results in a coarse PSD plot, as shown by the general fuzziness in the **Estimated PSD** plot.

In the **PSD (Window = 128)** graph, you can see that a small window generates a smooth **Estimated PSD** plot. However, a small window can lead to large bias, as you can see that the magnitudes of the peaks in the **Estimated PSD** plot are different from the magnitudes of the peaks in the **Ideal PSD** plot.

Use the TSA Welch VI to compute the PSD of a time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Bispectrum Estimation Methods

Bispectrum is a higher-order extension of power spectral estimation. A conventional power spectrum decomposes the power of a time series over frequency. In contrast, the bispectrum decomposes the third moment (skewness) of a time series over frequency. You can use the estimated bispectrum to detect asymmetric nonlinearities in a time series. Refer to the *Fault and Failure Diagnosis* section of Chapter 1, *Introduction to Time Series Analysis*, for information about a comparison between the PSD and bispectrum.

Use the TSA Bispectrum VI to compute the bispectrum of a time series. You can use this VI to compute the bispectrum based on the FFT or the AR model of the time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Cepstrum Estimation Methods

A *cepstrum*, which is an anagram of the word *spectrum*, is the Fourier transform of the natural logarithm of a spectrum. You can use the estimated cepstrum to identify echoes or periodic components in a time series. A cepstrum also is useful for separating homomorphic or convolved components in a time series by transforming the time series into a domain where the convolution becomes a simple summation operation. Cepstrum has three derivative forms—real cepstrum, complex cepstrum, and power cepstrum.

Estimating the Real Cepstrum of a Time Series

The real cepstrum is derived from the magnitude spectrum of a time series. You cannot reconstruct the original time series from the real cepstrum because the real cepstrum does not preserve the phase information of the original time series.

The real cepstrum is useful for feature extraction from homomorphic time series, such as the following applications:

- Speech analysis; specifically in formant and voice pitch tracking
- Machine fault diagnosis; specifically in gearbox and turbine fault detection

Use the TSA Real Cepstrum VI to compute the real cepstrum of a time series. As with power spectrum estimation, you can estimate the real cepstrum using the FFT or the AR model of the time series. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

Estimating the Complex Cepstrum of a Time Series

Unlike the real cepstrum, the complex cepstrum of a time series preserves the phase information of the original time series. Therefore, you can reconstruct the original time series from the power cepstrum. The complex cepstrum is useful in noise reduction, signal separation, and cepstral filtering.

Use the TSA Complex Cepstrum VI to compute the complex cepstrum of a time series. Use the TSA Inverse Complex Cepstrum VI to reconstruct a time series from the complex cepstrum. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about these VIs.

Figure 6-10 shows an example of removing echoes from a time series by processing the complex cepstrum. In this example, the original signal consists of three signal components: a decaying sinusoid starting at time 0, added with two distinct echoes of that signal later in time, as shown in the **Signal with Echoes** graph.

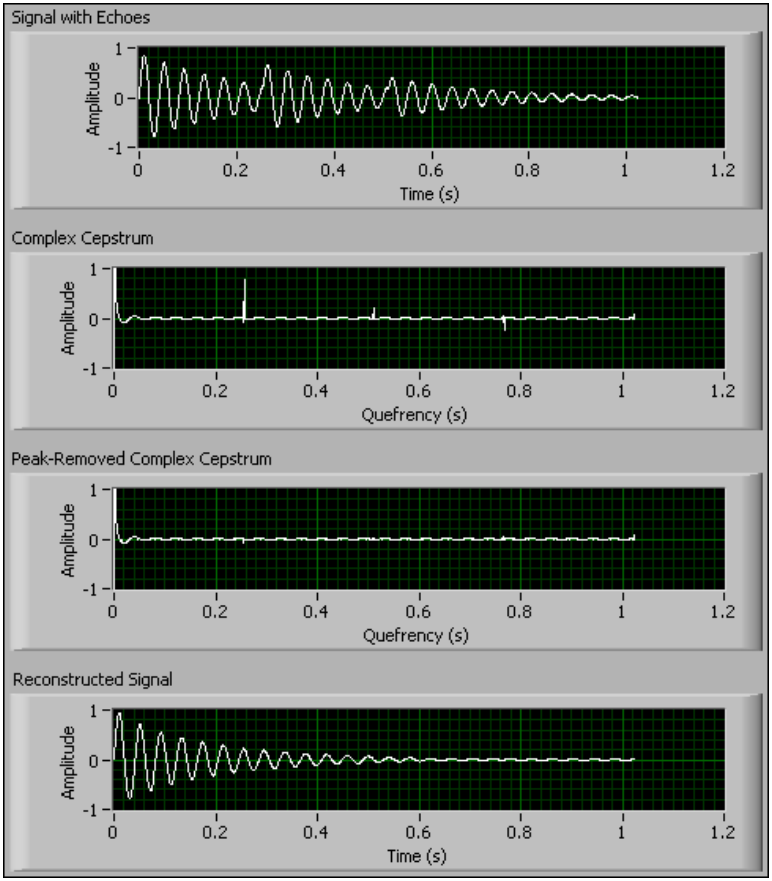

Figure 6-10. Removing Echoes from a Time Series

As Figure 6-10 shows, you can remove echoes from a time series using the following steps:

- Compute the complex cepstrum of the original time series using the TSA Complex Cepstrum VI. The peaks appearing in the **Complex Cepstrum** graph indicate the echo locations.
- Remove the peaks corresponding to the echoes from the complex cepstrum by setting the peak values to zero. The **Peak-Removed Complex Cepstrum** graph shows that the peaks have been removed.
- Reconstruct the original time series from the modified complex cepstrum using the TSA Inverse Complex Cepstrum VI. In the **Reconstructed Signal** graph, you can see that the reconstructed time series does not contain echoes.

Note In Figure 6-10, the word *quefrency* is an anagram of the word *frequency*.

Estimating the Power Cepstrum of a Time Series

The power cepstrum is an efficient tool for finding different harmonic families in the PSD of a time series. A power cepstrum is the inverse FFT transform of the natural logarithm of the PSD. You can compute the power cepstrum of a time series as follows:

$$C(\tau) = FFT^{-1}(\log(PSD))$$

The power cepstrum $C(\tau)$ is a real-valued time series.

Figure 6-11 shows the PSD and power cepstrum of a gearbox vibration signal, respectively.

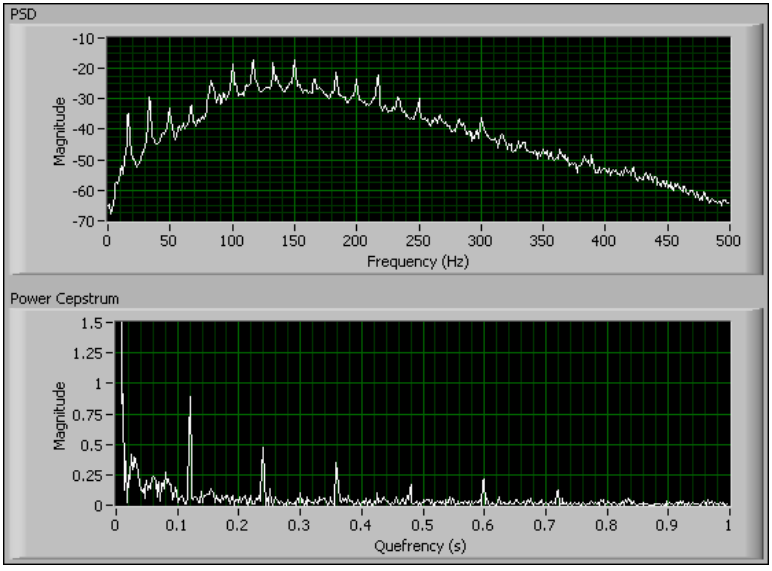

Figure 6-11. PSD and Power Cepstrum of a Gearbox Vibration Signal

The **PSD** graph suggests that the signal contains both periodic and non-periodic components. In the **Power Cepstrum** graph, you can see that the power cepstrum gives a more clear indication of harmonic peak families than the PSD.

You can modify, or lifter, the power cepstrum and then transform it back to the PSD. The word *lifter* is an anagram of the word *filter*, formed by reversing the first three letters. By liftering the unnecessary harmonic peak

families, you can remove an individual peak family from the PSD. Figure 6-12 shows the PSD of the gearbox vibration signal after you lifter all the harmonic peak families. Notice that the harmonic peaks in the liftered PSD have disappeared.

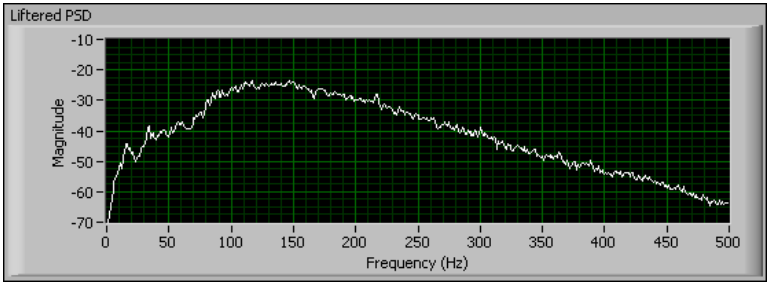

Figure 6-12. Liftered PSD of the Gearbox Vibration Signal

To lifter the harmonic families, complete the following steps:

1. Compute the power cepstrum of the vibration signal from the PSD.
2. Remove harmonic peaks of individual harmonic family that you want to discard from the power cepstrum.
3. Reconstruct the PSD from the liftered power cepstrum.

If a PSD contains several harmonic families, use the TSA Lifter PSD VI to separate harmonic peaks in the PSD by computing the power cepstrum and filtering the unnecessary harmonic peaks in the cepstrum. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about this VI.

A

References

This manual contains information about using the Time Series Analysis VIs in LabVIEW applications. In most cases, the underlying theory is omitted. If you are interested in the rigorous mathematical treatment of time series analysis techniques and algorithms, refer to the following textbooks and technical papers, which National Instruments used to develop the Time Series Analysis Tools.

- Chris Chatfield. 2004. *The Analysis of Time Series: An Introduction*. 6th edition. Chapman & Hall/CRC.
- Robert H. Shumway, David S. Stoffer. 2000. *Time Series Analysis and Its Applications*. Springer-Verlag. New York.
- *NIST/SEMATECH e-Handbook of Statistical Methods*. 2005. `http://www.itl.nist.gov/div898/handbook/`.
- Hua, Y., and T. K. Sarkar. 1990. "Matrix Pencil Method for Estimating Parameters of Exponentially Damped/Undamped Sinusoids in Noise." *IEEE Transaction on Acoustic, Speech, and Signal Processing* vol. 38.5: 814-824.
- A. Rivola, P.R. White. "Detecting System Non-linearities by Means of Higher Order Statistics."
- Aapo Hyvarinen, Erkki Oja. "Independent Component Analysis: Algorithms and Applications." Neural Networks Research Centre, Helsinki University of Technology
- B.Peeters and G. De Roech. 1999. "Reference-based Stochastic Subspace Identification for Output-Only Modal Analysis." *Mechanical Systems and Signal Processing* 13(6), 855-878.
- Ivan Goethals, Laurent Mevel, Albert Benveniste and Bart De Moor. "Recursive Output Only Subspace Identification for In-flight Flutter Monitoring."

B

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- **Support**—Online technical support resources at ni.com/support include the following:
  - **Self-Help Resources**—For answers and solutions, visit the award-winning National Instruments Web site for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
  - **Free Technical Support**—All registered users receive free Basic Service, which includes access to hundreds of Application Engineers worldwide in the NI Developer Exchange at ni.com/exchange. National Instruments Application Engineers make sure every question receives an answer.

    For information about other technical support options in your area, visit ni.com/services or contact your local office at ni.com/contact.
- **Training and Certification**—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.